# Machine Learning Anisotropic Coarse-Grained Simulation Models of Small-Molecule and Polymeric Organic Semiconductors

A THESIS PRESENTED
BY MARLTAN O. WILSON
TO
THE SCHOOL OF PHYSICS, CHEMISTRY AND EARTH SCIENCES IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN CHEMICAL SCIENCE

THE UNIVERSITY OF ADELAIDE
ADELAIDE, SOUTH AUSTRALIA

SUPERVISORS:

A/PROF. DAVID M. HUANG

A/PROF. TAK W. KEE

# Thesis Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree. I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Marltan Wilson
December 2022

## Abstract

A set of machine learning workflows have been developed to automate the generation of accurate anisotropic coarse-grained models and interaction potentials for small molecules and polymers as well as to analyze the aggregate structure of dilute semiflexible polymers with anisotropic monomers.

The multiscale coarse-graining method for isotropic coarse-grained particles has been extended to anisotropic coarse-graining of small molecules and polymers using a mixture of machine learning tools and classical simulation methods. The resulting coarse-grain interaction potentials derived from the machine-learned force-matching approach are flexible and scalable with respect to the type of molecules, the size of the simulation, and the simulation conditions. The robust deep-learning models were specifically used to construct coarse-grained interaction potentials for single-site anisotropic modeling of organic molecules and have shown the capability of reproducing the liquid crystal phase behavior of organic semiconductors.

An autoencoder machine learning approach has been used to automate the encoding of atomistic trajectories into unique anisotropic coarse-grained sites. This automated procedure allows for the creation of a simplified representation of organic polymers with the added feature of an accurate back mapping to the atomistic trajectories using the decoder network.

Machine learning tools are also developed in this work to analyze and predict the aggregation tendencies of small anisotropic molecules and organic semiconducting polymers in either the liquid or solution phase. A practical deep-learning framework, for the anisotropic coarse-graining of polymers and anisotropic macromolecules, was implemented alongside an automated workflow to predict the polymers' key aggregation behaviors based on their structure, flexibility, and the simulation condition.

# Acknowledgments

Firstly I want to acknowledge "the collective" for its contributions to every stage of this project. I want to acknowledge the expertise of and express gratitude to, the Huang–Kee research group many of whom contributed to the foundational work that makes this thesis possible. I must highlight my supervisor A/Prof Huang, who provided insight, advice, feedback, funding, and direction that has allowed me to complete this thesis. I am also thankful to the University of Adelaide for providing support and facilitating my Ph.D. journey.

Lastly, I want to highlight the intangible but invaluable light of a thousand summers, Chinatsu.

# Contents

# Chapter 1

# Introduction

## 1.1 Organic semiconductors

Organic semiconductors are historically constructed from carbon-based $\pi$-bonded molecules which have the ability to interact with visible light to generate charge carriers. These $\pi$-bonded molecules are typically aromatic rings with strong shape and interaction anisotropy, leading to directionally dependent charge transport. It is therefore widely understood that the morphology of the semiconducting materials affects their optoelectronic properties. Since their discovery, they have been used in a wide array of applications including, organic solar cells, organic light-emitting diodes, and biosensing. The proliferation of organic semiconductor technologies is due to their low cost, flexibility, and ease of production when compared to traditional inorganic semiconductors.[1] These organic semiconductors can be divided into two broad groups namely, polymers and small-molecules. The polymers can be constructed from monomers that are soluble in organic solvents allowing them to be solution-processed. On the other hand, small-molecules can be either soluble or insoluble in organic solvents and are generally thermally evaporated. As a result of these properties, there are many solution processing techniques that have been used to fabricate organic semiconducting materials such as ink-jet printing, reel-to-reel, and spin-coating. Organic semiconductors are thus seen as the future of mass-produced flexible electronic devices.[2] Even though polymer semiconductors have the potential to make tangible advances in next-generation technology, there are a few technical hurdles that must be overcome. Since solution processing and thermal deposition methods typically produce amorphous or polycrystalline material, one major obstacle is the lack of a theory that describes how aggregate structures in solution can be controlled to optimize optoelectronic properties.[3] Over the past decade, there have been numerous advances in refining a practical approach to achieving high-efficiency organic semiconductors. Alongside these advances, there has been an experimental exploration of a wide variety of polymers to elucidate the rules underpinning the most efficient acceptor-donor combinations.[4–9] However, until an empirical or theoretical model is developed to describe the control of aggregation, a purpose-driven design process of organic electronics will continue to be elusive.

## 1.2 Molecular dynamics simulation

The use of molecular dynamics software has become commonplace in the effort to understand the aggregation behavior of organic semiconductors.[10] Classical molecular dynamics refers to any computational method which uses Newtonian mechanics to simulate the interaction of atoms, molecules, or pseudo-atoms representing united groups of atoms. Bonds and angles are treated as classical springs and non-bonded van der Waals and electrostatic interactions are implemented using analytical pair-wise or many-body force-fields. Molecular dynamics simulations allow researchers to probe states that would otherwise be inaccessible to experimental techniques. Molecular dynamics simulations provide high resolution of molecules and finer control over experimental conditions than any experimental technique. In the case of organic semiconductors, It allows control over the starting configuration and provides a detailed description of the progression of the system. These finer controls are important for understanding the disorder and heterogeneity in organic semiconductor thin films and the overall non-equilibrium nature of film formation. Simulations also provide an opportunity to make observations while limiting variables that cannot be eliminated during experimentation.[11] Techniques involving simulations have been successfully applied to soft matter analysis in fields such as biology and engineering. However, these simulations have their limitations.

The most accurate results from molecular dynamics are obtained from all-atom fine-grained simulations. Under these conditions, each atom of a molecule is explicitly defined and tracked throughout the simulation. This process can lead to computationally expensive calculations. The major computational expense, in using molecular dynamics simulation, comes from the functional form of the non-bonded interactions. On the other hand, more complex non-bonded force fields usually result in more accurate simulations. The expense of these simulation processes scales with the number of degrees of freedom $(N)$, in some cases as $\mathscr{O}(N^3)$.[12] This computational cost places an upper bound on the size and the time scale of the simulation.[13] These time scales are often shorter than the time need to observe the aggregation dynamics of organic semiconductors. Currently, all-atom simulations or organic semiconductors can only access tens of nanometers and hundreds of nanoseconds). Even with advances in supercomputers and cheaper graphics processing units (GPUs), it is more practical to study the coarse-grained analog of the systems of interest.

## 1.3 Coarse-grained models

A coarse-grained model of a molecule such as the one shown in Fig 1.1 is one in which the individual atoms of the molecule are systematically mapped to a lower number of sites. Coarse-grained models can be developed based on reference to experimental data (top-down coarse-graining) or by referencing an underlying atomistic model (bottom-up coarse-graining). For the bottom-up method of coarse-graining, the map to coarse-grained sites is done to match the local structure and, sometimes, the local dynamics of the fine-grained atomistic model.[14–16] Multiscale coarse-graining[17,18] involving the iterative matching of forces acting on the coarse-grained site to the forces on the atomistic model, and iterative Boltzmann inversion[18,19] where the pair potential of particles is derived from the radial distribution function of the equilibrium ensemble are two methods of bottom-up coarse-graining while the statistical associating fluid theory,[18,20] which is an equation of state linking macroscopic thermodynamics such as densities and free energies to a molecular bead model, is an example of top-down coarse-graining. Currently, both bottom-up and top-down methods of coarse-graining have limitations such as representability, transferability, thermodynamic consistency as well as limited accuracy in reproducing the thermo-mechanical and dynamical prpperties of materials.

The process of coarse-graining reduces the total number of sites in a molecular dynamics simulation, which often leads to faster simulation or larger systems over longer time scales. Depending on the procedure used, coarse-graining can lead to speedups of two or three orders of magnitude.[21] These methods, however, are usually dependent on the experimenter's intuition about the system under consideration.[22] It is often the case that atoms in a polymer that are rigidly bonded together are coarse-grained into a single spherical site.[23] This procedure inevitably leads to information loss in the system. A major concern with the degree of coarse-graining is, how to determine the tolerance for information loss. There is also the problem of developing coarse-grain models that are representative of the state point at which the model is parameterized as well as coarse-grained models that are transferable to state points for which it was not parameterized.

There have been numerous papers looking at information loss resulting from the reduced degrees of freedom between the fine and coarse-grained models.[23,24] Information lost during coarse-graining can have significant implications for the analysis of the results obtained from coarse-grained molecular dynamics. Two such implications are the information lost about the flexural rigidity of molecules and $\pi$-stacking in molecules with aromatic backbones.[25] In organic semiconductors carrier mobility is highly directional, meaning that charge transport is fastest along the polymer backbone than along the $\pi$-stacking direction by up to two orders of magnitude.

Anisotropic coarse-grained models such as the one shown in Fig 1.2 that retain the general shape of each molecular subunit provide a better approximation to the fine-grain model than isotropic coarse-graining methods. However, there is a trade-off between the increased degrees of freedom and computational speed. On one hand, anisotropic models of molecules have been implemented classically, with great success, especially in biology where anisotropy plays a major role in the folding of protein chains.[26] On the other hand, there are very few cases where anisotropy has been implemented when considering organic semiconductors. These interactions are especially important because they determine the conformation of semiconducting polymer in solution. Multiscale coarse-graining (MS-CG) is a variational method of obtaining optimal coarse-grained potentials for use in coarse-grained molecular dynamics.[27] The coarse-graining equations for the MS-CG method come from enforcing thermodynamic consistency between the fine-grain and coarse-grain models, which results in a coarse-grained model that accurately captures the properties of the fine-grain model. The MS-CG method iteratively updates the parameters of a set of basis functions to optimize the effective force on each coarse-grained site by matching it to the force on the atomistic model.[22] The force-matching algorithm is straightforward, but, the choice of basis functions is open for manipulation.[28] Each choice of basis function comes with a set of variable parameters which approximates an optimal potential of mean force. The choice of basis functions also contributes to speedups based on the computational complexity.[29] Research is ongoing into the design and optimization of a set of computationally efficient basis functions with optimal tunability. The anisotropic force-matching coarse-graining (AFM-CG) method[30] is a bottom-up coarse-graining algorithm that extends the MS-CG method to anisotropic particles.

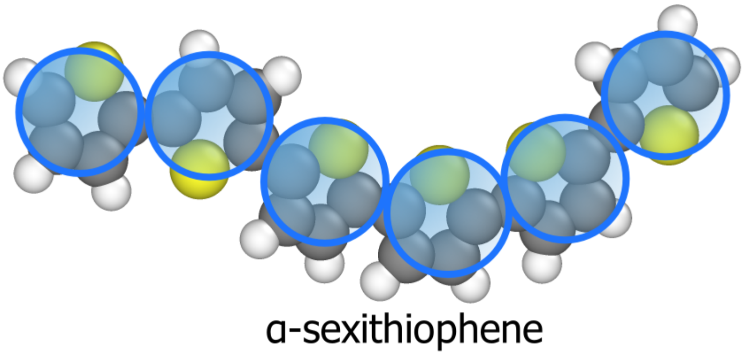


**Figure 1.1:** sexithiophen mapped to six isotropic coarse-grained sites.

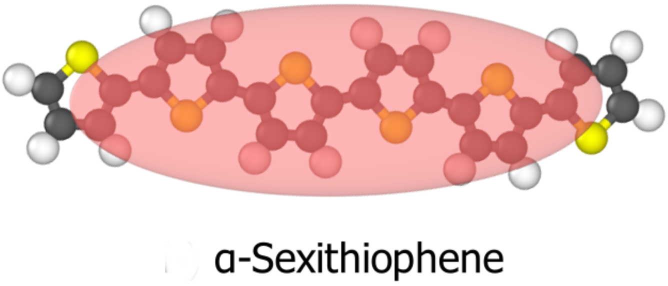


**Figure 1.2:** Sexithiophene mapped to a single anisotropic coarse-grained sites.

Other than the misaligned dynamics, there is also the issue of the lack of transferability of classically learned coarse-grained potentials. Potentials are usually dependent on the thermodynamic conditions under which they were designed and are generally not transferable.[10,31] Coarse-grained models usually have numerous fitting parameters needed to capture the structural distribution and dynamics of the fine-grain system. Machine learning has been explored as a possible solution to the optimization of these parameters. Even though bioinformatics polymer informatics is still a relatively new subdiscipline, rapid advances in machine learning technology have increased its popularity.[10] There has been significant effort to develop machine-learning force-fields that are transferable to state points beyond the initial parameterization. These approaches try to mitigate the limitations of pure bottom-up or top-down coarse-graining through a combination of both approaches.[32,33]

## 1.4 Machine learning

Machine learning is the term used to describe any algorithm that improves its performance on a task base on experience.[34] A typical machine learning algorithm consists of trainable parameters, input and target data used for training, validation data used to define a stopping criterion for the algorithm, and a loss metric that is used to measure the performance of the algorithm on learning the input data. Machine learning algorithms can be divided into two broad categories, supervised and unsupervised learning. Supervised learning provides an input data set along with a set of data labels. The prediction of the machine learning algorithm is then compared against the data labels using the loss metric to evaluate performance. On the other hand, unsupervised learning does not provide a set of labeled data, instead, the algorithm learns by a self-referential route. Unsupervised machine learning is primarily used for feature extraction and data compression. There are two major categories of algorithms used in machine learning to update trainable parameters, gradient, and non-gradient methods. Each of these categories is further divided into more specific families of algorithms such as evolutionary algorithms for non-gradient and neural networks for gradient methods. Evolutionary algorithms mimic evolutionary processes to find a set of solutions that best optimize a cost function subjected to a set of constraints. One of the major advantages of evolutionary algorithms is their ability to handle integer-valued functions which have no derivatives. Gradient-based learning algorithms are dependent on the back-propagation algorithm which calculates the derivative of the loss function with respect to all the trainable parameters in a neural network. The values of the trainable parameters are then iteratively updated based on the direction of the steepest descent on the loss surface. These neural network algorithms are useful for cases where the machine learning algorithm needs to produce differentiable functions.

The choice of machine learning algorithm used is highly dependent on the type of problem and information available. Artificial neural networks, as shown in Fig 1.3, are very good at solving regression and classification-type problems. Differentiability is an important factor when considering machine-learning applications for non-bonded interactions. The only limitation of these methods is the amount of data needed to build training and validation sets. Even though large volumes of data can be generated from molecular dynamics simulations, neural networks have not attracted enough attention with respect to the solution of the many problems in the development of anisotropic organic semiconductor coarse-grained models. Many machine-learning architectures have been deployed to solve some problems in coarse-grained molecular dynamics. Fully connected dense neural networks, for example, have been used to model the coarse-grained potential of mean force for spherical isotropic models. Potentials derived using machine learning can better capture the highly non-linear many-bodied interactions, when compared to classical coarse-graining methods.[35] These neural network potentials have also shown the meaningful ability to integrate seamlessly with molecular dynamics software. Another machine learning architecture that has been used in molecular dynamics is autoencoders. They are an example of a special type of neural network that has advanced data compression abilities. They are classified as an unsupervised machine learning model in that they learn without the need for a separately labeled data set. In an autoencoder, the input is passed from an input layer with $A$ nodes to a hidden layer with $B$ nodes, where $(A > B)$. The hidden layers are then connected to an output layer with $A$ nodes and the accuracy of the network is calculated as the reconstruction loss between the input and output layers. Recent developments in variational autoencoders have paved the way for their use as a means of coarse-graining.[36,37] The many variants of autoencoder along with the opportunity to develop specialized error functions and layers to optimize performance have attracted a lot of attention to this architecture.

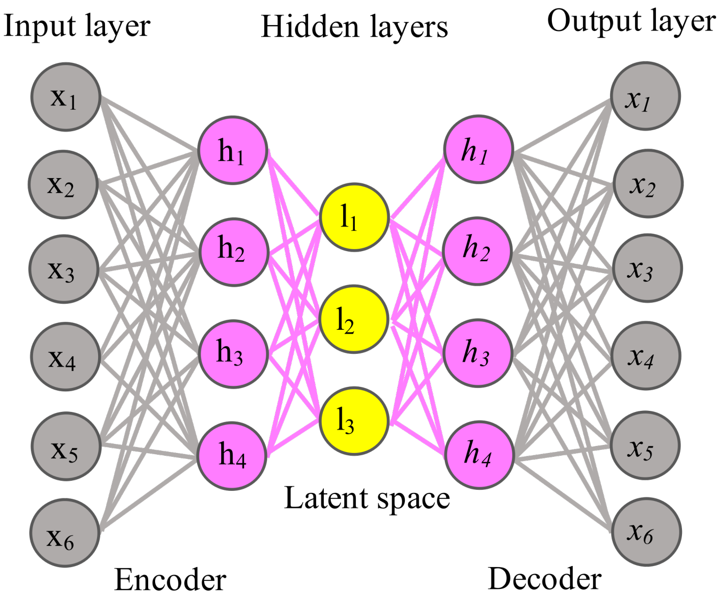


**Figure 1.3:** A schematic illustration of a typical feedforward artificial neural network showing the input, output, and hidden layers.

Polymer informatics is a fast-growing subdiscipline with plenty of opportunities to extend and modify existing methodologies. As discussed above, there are many areas of molecular dynamics where machine learning has been applied successfully. However, there has not been a robust deep-learning architecture applied to anisotropic coarse-graining in an attempt to recover both a configurational and thermodynamical optimized analog of the underlying fine-grain model. Where machine learning has been used to derive neural network potentials, the application has not been extended to the anisotropic model. There have been attempts to develop classical temperature-transferable coarse-grained potentials, but machine learning has not been applied to the problem. In cases where machine learning has been employed to solve any of the deficiencies, the solution is only given for simple monomers and as a result, many of the problems that are associated with modeling the aggregation of organic semiconductors in solution are not fully motivated. There have been efforts made to model the aggregation phase diagrams of dilute semiflexible isotropic polymer[38] as well as semiflexible polymers with anisotropic monomer units[39] using molecular dynamics simulations. However, this approach is limited by the size of the parameter space that can be effectively explored. A machine learning method capable of producing a reduced representation of a polymer aggregate that can then be clustered in a smooth latent space representation is more effective at extracting relationships between the polymer properties and the polymer aggregate. In the progress towards better machine-learning models there have been efforts to develop metrics for the evaluation of the transferability of these methods. The focus has shifted away from just comparing the model's energy and force losses to comparison of the materials predicted structures and properties.[40]

## 1.5 Thesis structure

This thesis consists of six chapters Chapter 1 provides a general introduction to the body of work highlighting the deficiencies in the field that are addressed in the following chapters. A brief overview of the most common methods and computational algorithms used in the preparation of this thesis is reviewed in chapter 2. Chapters 3–

5 are a collection of unpublished results written in publication format. Chapter 3 covers the introduction, design, and implementation of anisotropic high-dimensional neural network coarse-grained potentials used to model small molecules and organic semiconductors. Chapter 4 covers the implementation of a variational autoencoder used to predict the optimal number of coarse-grained sites, for anisotropic molecules and polymers to obtain the highest back-mapping fidelity. Chapter 5 addresses the design of a machine-learning workflow for the automatic labeling and prediction of aggregation phase space of semiflexible polymers with anisotropic monomers. The final chapter (Chapter 6) provides an overall conclusion to the body of work presented in the preceding chapters.

# References

[1] A. M. Bagher, Int. J. Renew. Sustain. Energy **3**, 53 (2014).

[2] W. Clemens, W. Fix, J. Ficker, A. Knobloch, and A. Ullmann, J. Mater. Res. **19**, 1963 (2004).

[3] Y. Liu, J. Zhao, Z. Li, C. Mu, W. Ma, H. Hu, K. Jiang, H. Lin, H. Ade, and H. Yan, Nat. Commun. **5**, 1 (2014).

[4] J. A. Bartelt, J. D. Douglas, W. R. Mateker, A. E. Labban, C. J. Tassone, M. F. Toney, J. M. Fréchet, P. M. Beaujuge, and M. D. McGehee, Advanced Energy Mater. **4**, 1301733 (2014).

[5] H. Lin, S. Chen, Z. Li, J. Y. L. Lai, G. Yang, T. McAfee, K. Jiang, Y. Li, Y. Liu, H. Hu, *et al.*, Advanced Mater. **27**, 7299 (2015).

[6] S.-O. Kim, D. S. Chung, H. Cha, M. C. Hwang, J.-W. Park, Y.-H. Kim, C. E. Park, and S.-K. Kwon, Solar Energy Mater. and Solar Cells **95**, 1678 (2011).

[7] P.-T. Wu, T. Bull, F. S. Kim, C. K. Luscombe, and S. A. Jenekhe, Macromol. **42**, 671 (2009).

[8] H. Hu, P. C. Chow, G. Zhang, T. Ma, J. Liu, G. Yang, and H. Yan, Acc. Chem. Res. **50**, 2519 (2017).

[9] L. Ye, S. Zhang, L. Huo, M. Zhang, and J. Hou, Acc. Chem. Res. **47**, 1595 (2014).

[10] T. E. Gartner III and A. Jayaraman, Macromol. **52**, 755 (2019).

[11] M. P. Allen, in *Compututation Soft Matter: from synthetic polymers to proteins , Vol. 23*, Vol. 23, edited by N. A. Kurt, John von Neumann Institute for Computing Jülich (John von Neumann Institute for Computing Jülich, Gustav-Stresemann-Institut, Bonn, Germany, 2004) Resreport 10, pp. 289–320, accessed 2023-05-22.

[12] A. Jain, N. Vaidehi, and G. Rodriguez, J. Comput. Phys. **106**, 258 (1993).

[13] S. Plimpton, Comput. Mater. Sci. **4**, 361 (1995).

[14] Y. Wang, W. Jiang, T. Yan, and G. A. Voth, Acc. Chem. Res. **40**, 1193 (2007).

[15] W. Tschöp, K. Kremer, J. Batoulis, T. Bürger, and O. Hahn, Acta Polym. **49**, 61 (1998).

[16] L. Lu, J. F. Dama, and G. A. Voth, J. Chem. Phys. **139**, 09B606_1 (2013).

[17] W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, J. Chem. Phys. **128**, 244114 (2008).

[18] T. D. Potter, J. Tasche, and M. R. Wilson, Phys. Chem.Chem. Phys. **21**, 1912 (2019).

[19] Z. Li, X. Bian, X. Yang, and G. E. Karniadakis, J. Chem. Phys. **145**, 044102 (2016).

[20] S. H. Huang and M. Radosz, Indus. Eng.Chem. Res. **29**, 2284 (1990).

[21] V. A. Harmandaris, D. Reith, N. F. Van der Vegt, and K. Kremer, Macromol. Chem. Phys. **208**, 2109 (2007).

[22] M. G. Saunders and G. A. Voth, Annu. Rev. Biophys. **42**, 73 (2013).

[23] S. O. Nielsen, C. F. Lopez, G. Srinivas, and M. L. Klein, J. Phys.: Condensed Matter **16**, R481 (2004).

[24] H. Wang, C. Junghans, and K. Kremer, Eu. Phys. E **28**, 221 (2009).

[25] T. S. Totton, A. J. Misquitta, and M. Kraft, J. Chem. Theory Comput. **6**, 683 (2010).

[26] E.-H. Yap, N. L. Fawzi, and T. Head-Gordon, Prote.: Struct. Func. and Bio. **70**, 626 (2008).

[27] F. Ercolessi and J. B. Adams, EPL **26**, 583 (1994).

[28] O. Akin-Ojo, Y. Song, and F. Wang, J. Chem. Phys. **129**, 064108 (2008).

[29] A. Das and H. C. Andersen, J.Chem. Phys. **131**, 034102 (2009).

[30] H. T. L. Nguyen and D. M. Huang, J. Chem. Phys. **156**, 184118 (2022).

[31] C. R. Ellis, J. F. Rudzinski, and W. G. Noid, Macromol.Theory and Sim. **20**, 478 (2011).

[32] Z. Shireen, H. Weeratunge, A. Menzel, A. W. Phillips, R. G. Larson, K. Smith-Miles, and E. Hajizadeh, npj Computational Materials **8**, 224 (2022).

[33] A. Giuntoli, N. K. Hansoge, A. van Beek, Z. Meng, W. Chen, and S. Keten, npj computational materials **7**, 168 (2021).

[34] D. M. Dutton and G. V. Conroy, Know. Eng. Rev. **12**, 341 (1997).

[35] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. De Fabritiis, F. Noé, and C. Clementi, ACS Cent. Sci. **5**, 755 (2019).

[36] K. K. Bejagam, S. Singh, Y. An, and S. A. Deshmukh, J. Phys. Chem. Let. **9**, 4667 (2018).

[37] W. Chen, A. R. Tan, and A. L. Ferguson, J. Chem. Phys. **149**, 072312 (2018).

[38] J. Zierenberg, M. Marenz, and W. Janke, Polymers **8**, 333 (2016).

[39] A. E. Cohen, N. E. Jackson, and J. J. De Pablo, Macromol. **54**, 3780 (2021).

[40] X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli, and T. Jaakkola, arXiv preprint arXiv:2210.07237 (2022).

# Chapter 2

# Theory and computational methods

## 2.1 Atomistic and coarse-grained simulations

Classical molecular dynamics algorithms[1] execute simulations in which the time evolution of the particles in a system is updated according to Newton's equations for a specified potential energy function for atoms or finite-size particles as shown in Fig 2.1. The force on a particle $i$ at position $\boldsymbol{r}_i$ and mass $m_i$ is the negative derivative of the interaction potential energy given as,

$$\boldsymbol{f}_i(\boldsymbol{r}^n) = -\frac{\partial U}{\partial \boldsymbol{r}_i} \tag{2.1}$$

for a system with $n$ particles. According to Newton's second law, the force on a classical particle is related to its mass and acceleration since,

$$\boldsymbol{f}_i(\boldsymbol{r}^n) = m_i \frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2} \tag{2.2}$$

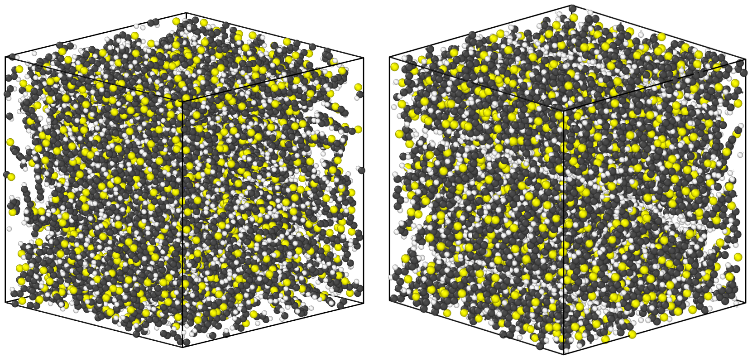
and the second derivative of position with respect to time $t$ is the acceleration of particle $i$. Additionally, for finite-size particles with center-of-mass position,

$$\boldsymbol{R}_I = \frac{\sum_{i \in \zeta_I} m_i \boldsymbol{r}_i}{\sum_{i \in \zeta_I} m_i}, \tag{2.3}$$

the rotation-inducing torque is calculated using,

$$\boldsymbol{\tau}_I(\boldsymbol{R}^N, \boldsymbol{\Omega}^N) = -\sum_q \boldsymbol{\Omega}_{I,q} \times \frac{\partial U}{\partial \boldsymbol{\Omega}_{I,q}} \tag{2.4}$$

where $\boldsymbol{\Omega}_I$ is the orientation of finite-size particle $I$.



(a) Isotropic sexithiophene

(b) Crystalline sexithiophene

**Figure 2.1:** Snapshots from a molecular dynamics simulation of (a) isotropic sexithiophene and (b) crystalline sexithiophene.

The interaction potential $U$ is usually taken as a sum over different contributions which are parameterized to reproduce the physical observables of a system at a particular state point. That is,

$$U = U_{\text{vdw}} + U_{\text{coulombic}} + U_{\text{bonds}} + U_{\text{angle}} + U_{\text{dihedral}} \tag{2.5}$$

where $U_{\text{vdw}}$ and $U_{\text{coulombic}}$ are the van der Waals and coulombic non-bonded interactions, $U_{\text{bonds}}$, $U_{\text{angle}}$, and $U_{\text{dihedral}}$ are interactions due to particles being separated by one or more covalent bonds. These interactions are typically approximated using analytical functions such as

$$U_{\text{coulombic}} = \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{r_{ij}}, \tag{2.6}$$

and the van der Waals forces can be calculated from isotropic potentials such as the Lennard-Jones potential given as,

$$U_{\text{LJ}} = 4\varepsilon_{ij}\left(\alpha_{ij}^{12} - \alpha_{ij}^{6}\right) \tag{2.7}$$

$\varepsilon_{ij}$ is the depth of the potential well and $\alpha_{ij} = \sigma_{ij}/r_{ij}$, where $\sigma_{ij}$ is the distance where the particle-particle potential is zero. For finite-size ellipsoidal particles, the van der Waals forces can be calculated from anisotropic potentials such as the Gay-Berne potential given as,

$$\begin{aligned} U_{\text{GB}} &= U_r(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{r}_{12}, \gamma) \cdot \eta_{ij}(\boldsymbol{\Omega}_i, \boldsymbol{\Omega}_j, \nu) \cdot \\ &\quad \chi_{ij}(\boldsymbol{\Omega}_i, \boldsymbol{\Omega}_j, \boldsymbol{r}_{ij}, \mu) \end{aligned} \tag{2.8}$$

where, $\boldsymbol{\Omega}_i, \boldsymbol{\Omega}_j$ are transformation matrices from the simulation box frame to body frame, $U_r$ is the shifted distance-dependent interaction based on the distance of closest approach and user-defined parameter $\gamma$ $\eta_{ij}$ and $\chi_{ij}$ are orientation-dependent energies based on the user-specified values of $\nu$ and $\mu$ respectively.

$$U_{\text{bond}} = K_{\text{B}}(b - b_0)^2 \tag{2.9}$$

where $K_\mathrm{B}$ is the bond stretching coefficient, and

$$U_\mathrm{angle} = K_\mathrm{A}(\theta - \theta_0)^2 \tag{2.10}$$

where $K_\mathrm{A}$ is the angle coefficient, and

$$\begin{aligned} U_\mathrm{dihedral} &= \frac{1}{2}K_1[1 + cos(\phi)] + \frac{1}{2}K_\mathrm{D}[1 - cos(2\phi)] \\ &+ \frac{1}{2}K_3[1 + cos(3\phi)] + \frac{1}{2}K_4[1 - cos(4\phi)] \end{aligned} \tag{2.11}$$

where $\phi$ is the dihedral angle and $K_\mathrm{D}$ is a fit parameter.

There is a myriad of possible molecular dynamics packages[2–6] available but all atomistic simulations in this thesis are done with the LAMMPS[7,8] simulation package using the OPLS-AA force fields,[9,10] using either the constant pressure or constant volume ensemble unless otherwise stated. The OPLS-AA force field was specifically developed to describe liquid-phase organic molecules. It has been shown to perform well in reproducing the flexibility, bulk liquid structure, and energetics of organic molecules. By using these previously optimized molecular models as a starting point, precise coarse-grained models can be parameterized.

A simulation box with periodic boundary conditions is used for the simulation of bulk liquids. Under these conditions, the simulation box is replicated in all directions infinitely ensuring that particles that move beyond the boundary of the simulation box are not lost. The minimum image convention is then used to calculate the non-bonded and short-ranged interactions between particles, where interactions are calculated over the shortest distance between particle images. This procedure ensures that bulk properties can be obtained without surface or finite-size effects while also reducing the total number of interactions that need to be calculated. A particle–particle particle–mesh method is used to calculate long-ranged electrostatic forces by mapping charges to a mesh in 3D space. The simulations are usually done using either the constant pressure or constant volume ensemble unless otherwise stated. In the constant pressure ensemble, the number of particles, the pressure, and, the temperature are fixed. While the volume, temperature, and number of particles are fixed for the constant volume ensemble. To maintain the pressure and temperature of these simulations, a Nosé-Hoover thermostat and barostat is used where additional fictitious variables are used to produce energy and volume fluctuations.

The simulations used for the parameterization of coarse-grained models are typically done in the isotropic liquid phase or solution phase. The AFM-CG method[11] is used for the force, torque, and virial matching procedure which ensures thermodynamic consistency in the configuration and momentum space as well as improves temperature transferability. The force, torque, and virial matching conditions are such that,

$$\boldsymbol{F}_I(\boldsymbol{R}^N, \boldsymbol{\Omega}^N) = -\frac{\partial U}{\partial R_I} = \left\langle \sum_{i \in \zeta_I} \boldsymbol{f}_i \right\rangle_{R^N, \Omega^N} \tag{2.12}$$

where $F_I$ is the coarse-grained force and

$$\boldsymbol{\tau}_I(\boldsymbol{R}^N, \boldsymbol{\Omega}^N) = -\sum_q \Omega_{I,q} \times \frac{\partial U}{\partial \Omega_{I,q}} = \left\langle \sum_{i \in \zeta_I} \Delta \boldsymbol{r}_i \times \boldsymbol{f}_i \right\rangle_{R^N, \Omega^N}, \tag{2.13}$$

where $\tau_I$ is the coarse-grained torque and

$$\begin{aligned} W(\boldsymbol{R}^N, \Omega^N, V) &= -\frac{\partial U}{\partial V} \\ &= \left\langle \frac{(n-N)k_{\mathrm{B}}T}{v} + \frac{1}{3v}\sum_{i=1}^{n} \boldsymbol{f}_i \cdot \boldsymbol{r}_i \right\rangle_{R^N, \Omega^N, V} \end{aligned} \tag{2.14}$$

where $W$ is the coarse-grained virial and $V$ is the corresponding system volume.

This choice of parameterization condition ensures that the coarse-grained potentials have a volume or density dependence[12,13] thus improving the temperature transferability[14] of the coarse-grained model. This temperature transferability is especially important for molecules such as sexithiophene which exhibit multiple liquid crystal phases[15] between the crystalline and isotropic phases. One way of assessing the temperature transferability of the coarse-grained potential is through structural comparison between atomistic and coarse-grained models of the simulated material at different temperatures. Structure indicators such as the angular-radial distribution function (ARDF) can be used to measure the accuracy of the coarse-grained model at different temperatures, where the ARDF[16] defined by

$$g(r,\theta) = \frac{\langle n(r,\theta)\rangle}{\frac{4}{3}\pi\rho\,[(r+\Delta r)^3 - r^3]\sin\theta\Delta\theta}, \tag{2.15}$$

where $\langle n(r,\theta)\rangle$ is the average number of molecules in the spherical shell within the bounds $r$ to $r+\Delta r$ of the center-of-mass of a chosen molecule and having an out-of-plane axis rotation of $\theta$ with respect to the out-of-plane axis of the chosen molecule and $\rho$ is the bulk number density. To further confirm that the density changes were associated with transitions from crystalline through nematic and smectic to the isotropic phase, the scalar orientational order parameter $P_2$ can be used. For a given simulation snapshot at time $t$, $P_2$ can be found by diagonalizing the ordering matrix $\boldsymbol{Q}$,

$$\boldsymbol{Q}(t) = \frac{1}{2N}\sum_{I=1}^{N}[3\boldsymbol{u}_I(t)\otimes\boldsymbol{u}_I(t) - \boldsymbol{E}], \tag{2.16}$$

where $\boldsymbol{u}_I$ is the unit vector along the molecular axis and $\boldsymbol{E}$ is the identity matrix. $\langle P_2\rangle$ is the average over the largest eigenvalue of this matrix for all snapshots of equilibrium configurations. When the ARDF is used alongside the scalar orientational order parameter,[17,18] a differentiation can be made between the liquid crystal phases of an organic semiconductor. The coarse-grained simulations are typically done to mirror the conditions of the atomistic simulations. It is then easier to evaluate the performance of the coarse-grained model. In cases where only coarse-grain simulations are done without the corresponding fine-grain simulation, the models are usually representative of typical semiconducting polymers.

## 2.2 Machine learning

Neural networks are one type of machine-learning algorithm based on mimicking a real neuron's function. The simplest neural network is the feedforward neural network shown in Fig 2.2. A feedforward neural network typically takes in an input $\boldsymbol{x}$ applies some weight $\boldsymbol{C}$ and bias $b$ followed by an activation function $f$ to produce an output $g$. That is, the neural network output is given by the following equation:

$$g = f(\boldsymbol{C} \cdot \boldsymbol{x} + b) \tag{2.17}$$

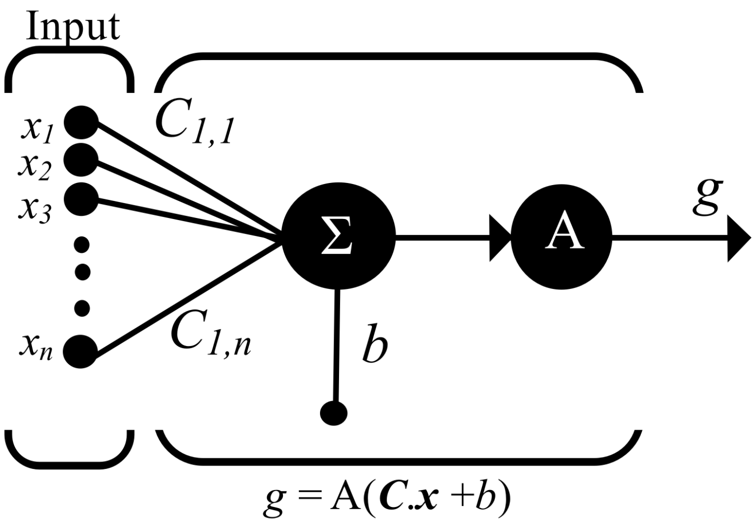


**Figure 2.2:** Schematic illustration of the internal working of a feedforward neural network with input ($\boldsymbol{x}$), weight matrix ($\boldsymbol{C}$), bias ($b$), activation function (A) and output ($g$)

Multiple layers can be used to model very complex multidimensional nonlinear functions such as the coarse-grained potential of mean force. The network learns by calculating the gradient of an error function with respect to the neural network parameters and using backpropagation to update the weight and biases of all the layers.[19] Backpropagation refers to the algorithm which calculates the derivative of the neural network parameters with respect to the loss. Derivatives are calculated using the chain rule iteratively starting from the last layer to avoid redundant calculations. A gradient descent algorithm is then used to update the parameters in the direction of steepest descent with respect to the gradient. The size of the loss that is used to update each neural network parameter is scaled by the learning rate in order to prevent overshooting the global minimum of the loss surface. Many of the new gradient descent algorithms have built-in adaptive learning rates for efficient convergence to the global minimum. The Tensorflow package developed by Google[20] and Keras libraries enable the rapid implementation and testing of machine learning models.[21] Tensorflow and Keras already have the built-in infrastructure to execute backpropagation, multiple gradient descent algorithms, and a wide variety of optimizers and loss functions. These libraries can be imported and used in python and have one of the largest developer communities.

The construction of machine-learned neural network potentials requires the neural network activation function to be continuous and differentiable. Unlike the case for deep learning, the tanh activation function is sufficient for this application since there is a limited chance of finding a system that could lead to vanishing gradients in the training of the neural network. Using a feedforward neural network to implement the force matching between the atomistic and coarse-grained model means that the output of the neural network function should be the potential whose derivative with respect to the coarse-grain positions should produce the forces on the coarse-grained sites. Implementation of this architecture is enabled in TensorFlow by the Gradient Tape algorithm. Gradient Tape is the algorithm that allows the calculation and storage of the derivatives of neural network parameters with respect to any other connected neural network parameter. Through this method, higher derivatives or multiple derivatives can be calculated during the forward propagation of data through the network. It uses the same mechanics as the backpropagation algorithm to calculate derivatives using the chain rule. This algorithm also facilitates more advanced matching conditions for anisotropic particles such as torque and virial matching.

More advanced neural network architectures are also possible within TensorFlow, including generic feed-forward neural networks, long short-term memory networks, and variational autoencoders.[22] Autoencoders in general are particularly useful for data compression and classification.[23] Since they are a form of unsupervised learning algorithm[24] their output is the same as the input, which makes them ideal for the development of back-mapping algorithms or the identification and classification of polymer aggregates[25] in large unlabeled data sets. Other networks such as feed-forward neural networks are ideal for implementing force-matching conditions since they produce smooth functions and the complexity of the model can be increased through the implementation of additional traditional layers or custom layers with filters or symmetry functions. Symmetry functions are especially important in the representation of atomic or molecular environments because they preserve the inherent symmetries of the underlying potential. These symmetry functions enforce the cut-off radius for non-bonded interactions as well as wrap angular coordinates to remove coordinate-induced singularities from molecular orientation representation. A combination of supervised and unsupervised algorithms can be used effectively to build workflows to study very complex systems. This is possible because autoencoders can generate a latent space which is optimal for feature extraction. These latent space representation are important when there is a desire to generate generalized description of complex systems.

# References

[1] A. T. Celebi, S. H. Jamali, A. Bardow, T. J. Vlugt, and O. A. Moultos, Mol. Sim. **47**, 831 (2021).

[2] A. Nurisso, A. Daina, and R. C. Walker, Homo. mod. , 137 (2011).

[3] J. A. Anderson, J. Glaser, and S. C. Glotzer, Comput. Mater. Sci. **173**, 109363 (2020).

[4] S. Páll, M. J. Abraham, C. Kutzner, B. Hess, and E. Lindahl, in *International conference on exascale applications and software* (Springer, 2014) pp. 3–27.

[5] C. Hu, H. Bai, X. He, B. Zhang, N. Nie, X. Wang, and Y. Ren, Comput. Phys. Commun. **211**, 73 (2017).

[6] A. Shkurti, M. Orsi, E. Macii, E. Ficarra, and A. Acquaviva, J. Comput. Chem. **34**, 803 (2013).

[7] W. M. Brown, P. Wang, S. J. Plimpton, and A. N. Tharrington, Comput. Phys. Commun. **182**, 898 (2011).

[8] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington, Comput. Phys. Commun. **183**, 449 (2012).

[9] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

[10] W. L. Jorgensen and N. A. McDonald, J. Mol. Struct.: THEOCHEM **424**, 145 (1998).

[11] H. T. L. Nguyen and D. M. Huang, J. Chem. Phys. **156**, 184118 (2022).

[12] A. Das and H. C. Andersen, J. Chem. Phys. **132**, 164106 (2010).

[13] S. Izvekov, P. W. Chung, and B. M. Rice, J. Chem. Phys. **133**, 064109 (2010).

[14] D. D. Hsu, W. Xia, S. G. Arturo, and S. Keten, Macromol. **48**, 3057 (2015).

[15] W. Bu, H. Gao, X. Tan, X. Dong, X. Cheng, M. Prehm, and C. Tschierske, Chem. Commun. **49**, 1756 (2013).

[16] S. Lorenz, T. R. Walsh, and A. Sutton, J. Chem. Phys. **119**, 2903 (2003).

[17] Y. A. Nastishin, H. Liu, T. Schneider, V. Nazarenko, R. Vasyuta, S. Shiyanovskii, and O. Lavrentovich, Phys. Rev. E **72**, 041711 (2005).

[18] A. Pizzirusso, M. Savini, L. Muccioli, and C. Zannoni, J. Mater. Chem. **21**, 125 (2011).

[19] R. Rojas, in *Neural networks* (Springer, 1996) pp. 149–182.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, in *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (2016) pp. 265–283.

[21] F. Chollet, *Deep learning with Python*, edited by M. P. Co. (Simon and Schuster, 2021).

[22] D. P. Kingma, M. Welling, *et al.*, Found. Trends Mach. Learn. **12**, 307 (2019).

[23] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. De Fabritiis, F. Noé, and C. Clementi, ACS Cent. Sci. **5**, 755 (2019).

[24] P. Baldi, in *Proceedings of ICML workshop on unsupervised and transfer learning* (JMLR Workshop and Conference Proceedings, 2012) pp. 37–49.

[25] J. Zierenberg, M. Marenz, and W. Janke, Polymers **8**, 333 (2016).

# Chapter 3

# Anisotropic molecular coarse-graining by force and torque matching with neural networks

## 3.1 Abstract

We develop a machine-learning method for coarse-graining condensed-phase molecular systems using anisotropic particles. The method extends currently available high-dimensional neural network potentials by addressing molecular anisotropy. We demonstrate the flexibility of the method by parametrizing single-site coarse-grained models of a rigid small molecule (benzene) and a semi-flexible organic semiconductor (sexithiophene), attaining structural accuracy close to the all-atom models for both molecules. The machine-learning method of constructing the coarse-grained potential is shown to be straightforward and sufficiently robust to capture anisotropic interactions and many-body effects. The method is validated through its ability to reproduce the structural properties of the small molecule's condensed phase and the phase transitions in the semi-flexible molecule over a wide temperature range.

## 3.2 Introduction

Machine learning is quickly becoming an invaluable tool in the search, analysis, and development of new materials.[1,2] Neural networks, in particular, have had major recent success in areas ranging from predicting the folding geometry of biological macromolecules such as proteins[3] to developing highly accurate temperature-transferable interatomic potentials.[4,5]

The latter is an important advance in the field of molecular dynamics (MD) simulations. Improvements in these machine-learning models aim to expand the length and time scale of simulations without sacrificing accuracy.[6,7] Currently used ab initio molecular dynamics simulation models are generally accurate but are computationally expensive, limiting their ability to probe long time scales.[8,9] However, neural-network potentials can produce ab initio accuracy at the computational cost of classical atomistic models.[10,11]

Even though simulations at the classical MD level are faster than ab initio MD, the speedup is still insufficient to model the long time scales needed to fully understand certain phenomena and processes such as supramolecular assembly. It is well known that explicit modeling of high-frequency motion is not critical for describing many phenomena in molecular systems. These simplifications have led to the development of molecular coarse-grained models to study large, complex materials and biological systems.[12] Parameterization of coarse-grained interaction potentials commonly takes one of two approaches: the top-down approach in which parameters are tuned to match macroscopic observables, as exemplified by the Martini model,[13] and the bottom-up approach in which interactions are derived from the properties of a fine-grained model with more degrees of freedom.[12] By following a similar bottom-up process used to apply machine learning to ab initio MD data, neural-network approaches have been extended to coarse-grained molecular models, further extending the length and time scale of simulations with atomistic accuracy.[14,15]

Neural-network potentials using isotropic coarse-grained particles have several advantages over their pair-wise additive analytical counterparts since they are constructed as many-body potentials. This many-body potential can become costly when multiple coarse-grained particles are needed to preserve the shape anisotropy. It is sometimes more accurate and computationally efficient to represent these groups of atoms as a single anisotropic coarse-grained particle such as an ellipsoid, such as in the case of large, rigid, anisotropic molecular fragments. Analytical anisotropic coarse-grained potentials such as the Gay-Berne potential[16,17] were developed to address the poor performance of spherically symmetric potentials in replicating intrinsic anisotropic interactions such as $\pi$-stacking. By modeling rigid anisotropic groups of atoms as ellipsoids, the anisotropic properties of the group are preserved in a single-site model. Shape and interaction anisotropy is especially important for the study of organic semiconductor molecules, which typically consist of highly anisotropic and rigid $\pi$-conjugated units and often form liquid-crystal phases whose morphology strongly affects their performance in devices such as solar cells, transistors, and light-emitting diodes.[18]

Unlike analytical pair-wise additive potentials such as the Gay-Berne potential, high-dimensional neural-network potentials are constructed based on the immediate neighborhood of a molecule and thus account for many-body effects as well as local density variations. Notable machine-learning implementations of inter-atomic and inter-molecular potentials include the neural-network potentials developed by Behler et al.[19] The Behler neural-network potentials are constructed from a set of symmetry functions used to represent the invariant properties of the atomic environment of each atom taken from ab initio simulations. DeepMD[10] and DeepCG[14] are two other neural-network codes constructed for atomistic and coarse-grained simulations, respectively. All of these neural-network potentials rely on an invariant representation of the atomic/molecular environment. The CGnets deep-learning approach[15] employed a prior potential to account for areas in a coarse-grained data set that may not be properly sampled due to high repulsive energies. These interactions are especially important to reproduce the local structure of the simulated material.

Machine learning has previously been applied to the parameterization of coarse-grained models with anisotropic particles,[20] but no such implementation has used a nonlinear neural-network optimization method to construct the coarse-grained potential. In this work, we address this gap in knowledge by using a neural network to construct a high-dimensional anisotropic coarse-grained potential. We parameterize the neural-network potential using a recently derived systematic and general bottom-up coarse-graining method called anisotropic force-matching coarse-graining (AFM-CG)[21] which generalizes the multi-scale coarse-graining (MS-CG) method[22] for isotropic coarse-grained particles to anisotropic particles. The method rigorously accounts for finite-temperature, many-body effects without assuming a specific functional form of the anisotropic coarse-grained potential. It yields general equations relating the forces, torques, masses, and moments of inertia of the coarse-grained particles to properties of a fine-grained (e.g. all-atom) molecular dynamics simulation based on a mapping between fine-grained and coarse-grained coordinates and momenta, and by matching the equilibrium coarse-grained phase-space distribution with the mapped distribution of the fine-grained system. The previous implementations of the AFM-CG method approximated the coarse-grained potential as a sum of pair interactions between particles.[21] Here, we extend this approach to more general many-body anisotropic interactions described by a neural network potential. We also extend the approach, which was derived for constant-volume systems in the canonical ensemble to constant-pressure systems by applying a virial-matching condition previously derived for the MS-CG method.

A general coarse-grained potential should capture any temperature-dependent phase transitions associated with either melting, annealing, or glass transition temperatures as well as the local structure and density of the material. The focus is on the development of a model for which trained parameters can be easily obtained and one capable of reproducing interaction anisotropy, temperature transferability, and many-body effects. The flexibility of the new model is demonstrated through the matching of structural and thermodynamic properties of condensed-phase systems of a small anisotropic molecule, benzene, and of a larger, more flexible organic semiconductor molecule, sexithiophene. These two molecules were chosen to determine the conditions under which coarse-grained structural inaccuracy outweighs the computational efficiency of a single-anisotropic-site model.

## 3.3 Theory

The key aspects of the theory that underpins the AFM-CG method and its extension to constant pressure via virial matching are summarized below. The reader is referred to Ref. 21 for a more detailed description of the AFM-CG method and the full derivation of its equations.

The positions $\boldsymbol{r}^n = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_n\}$ of the $n$ fine-grained particles are mapped onto the positions $\boldsymbol{R}^N = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_N\}$ and orientations $\boldsymbol{\Omega}^N = \{\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \ldots, \boldsymbol{\Omega}_N\}$ of the $N$ anisotropic coarse-grained particles. Each fine-grained particle $i$ is mapped to a single coarse-grained particle by defining $N$ non-intersecting subsets, $\zeta_1, \zeta_2, \ldots, \zeta_N$, of the FG particle indices such that $\zeta_I$ contains the indices of fine-grained particles mapped onto coarse-grained particle $I$. The position $\boldsymbol{R}_I$ of coarse-grained particle $I$ is defined to be equal to the center-of-mass of the group of FG particles that are mapped onto it, i.e.

$$\boldsymbol{R}_I = \frac{\sum_{i \in \zeta_I} m_i \boldsymbol{r}_i}{\sum_{i \in \zeta_I} m_i}, \tag{3.1}$$

where $m_i$ is the mass of FG particle $i$. The orientation

$$\boldsymbol{\Omega}_I = \begin{bmatrix} \boldsymbol{\Omega}_{I,1} \\ \boldsymbol{\Omega}_{I,2} \\ \boldsymbol{\Omega}_{I,3} \end{bmatrix} \tag{3.2}$$

of coarse-grained particle $I$ is specified by the rotation matrix whose components are the particle's three normalized principal axes of inertia, $\boldsymbol{\Omega}_{I,q}$ for $q = 1,2,3$. These axes are defined to be equal to the corresponding principal axes relative to the center-of-mass of the group of fine-grained particles that are mapped onto the coarse-grained particle. Thus, these axes are the normalized eigenvectors of the inertia tensor

$$\mathbb{I}_{\mathrm{FG},I} = \sum_{i \in \zeta_I} m_i (||\Delta \boldsymbol{r}_i||^2 \boldsymbol{E} - \Delta \boldsymbol{r}_i \Delta \boldsymbol{r}_i^{\mathrm{T}}), \tag{3.3}$$

where $\Delta \boldsymbol{r}_i = \boldsymbol{r}_i - \boldsymbol{R}_I$ is the position of fine-grained particle $i$ relative to the center-of-mass (coarse-grained particle position) and $\boldsymbol{E}$ is the $3 \times 3$ identity matrix. From these coordinate mappings and the relationship between generalized coordinates and momenta from Hamilton's equations,[23] mappings from the linear momenta $\boldsymbol{p}^n = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ of the fine-grained particles to the linear momenta $\boldsymbol{P}^N = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_N\}$ and angular momenta $\boldsymbol{L}^N = \{\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_N\}$ of the anisotropic coarse-grained particles can also be defined.[21] The mappings for coarse-grained particle $I$ are

$$\boldsymbol{P}_I = \frac{M_I}{\sum_{i \in \zeta_I} m_i} \sum_{i \in \zeta_I} \boldsymbol{p}_i \tag{3.4}$$

and

$$\boldsymbol{L}_I = \mathbb{I}_I \mathbb{I}_{\mathrm{FG},I}^{-1} \sum_{i \in \zeta_I} \Delta \boldsymbol{r}_i \times \boldsymbol{p}_i, \tag{3.5}$$

respectively, where $\mathbb{I}_I$ is the inertia tensor of coarse-grained particle $I$.

Given these mappings, several conditions can be derived that the coarse-grained model must satisfy for its equilibrium coarse-grained phase-space distribution to match the corresponding mapped distribution of the fine-grained system. Consistency between the configuration-space distributions gives the following matching conditions between the forces $\boldsymbol{F}_I$ and torques $\boldsymbol{\tau}_I$ on coarse-grained particle $I$ and the forces on the fine-grained particles mapped onto it:[21]

$$\boldsymbol{F}_I(\boldsymbol{R}^N, \boldsymbol{\Omega}^N) = -\frac{\partial U}{\partial R_I} = \left\langle \sum_{i \in \zeta_I} \boldsymbol{f}_i \right\rangle_{R^N, \Omega^N} \tag{3.6}$$

and

$$\boldsymbol{\tau}_I(\boldsymbol{R}^N, \boldsymbol{\Omega}^N) = -\sum_q \Omega_{I,q} \times \frac{\partial U}{\partial \Omega_{I,q}} = \left\langle \sum_{i \in \zeta_I} \Delta \boldsymbol{r}_i \times \boldsymbol{f}_i \right\rangle_{R^N, \Omega^N}, \tag{3.7}$$

where $U(R^N, \Omega^N)$ is the coarse-grained potential, $\boldsymbol{f}_i(\boldsymbol{r}^n) = -\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{r}_i}$ is the force on fine-grained particle $i$, with $\boldsymbol{u}(\boldsymbol{r}^n)$ the fine-grained potential and $\langle \cdots \rangle_{R^N, \Omega^N}$ denoting an average over fined-grained configurations mapped to coarse-grained configuration $(R^N, \Omega^N)$.

Consistency between the momentum-space distributions requires the mass $M_I$ of coarse-grained particle $I$ to be the sum of the masses of its constituent fine-grained particles, i.e.[21]

$$M_I = \sum_{i \in \zeta_I} m_i. \tag{3.8}$$

In addition, provided that the inertia tensor $\mathbb{I}_{\mathrm{FG},I}$ of the group of fine-grained particles mapped to this coarse-grained particle does not depend on the configuration of the other particles,[21]

$$I_{I,q}^{1/2} \exp\left(-\frac{I_{I,q}\omega_{I,q}^2}{2k_{\mathrm{B}}T}\right) \approx \left\langle I_{\mathrm{FG},I,q}^{1/2} \exp\left(-\frac{I_{\mathrm{FG},I,q}\omega_{I,q}^2}{2k_{\mathrm{B}}T}\right) \right\rangle_{\boldsymbol{R}_I, \boldsymbol{\Omega}_I}, \tag{3.9}$$

where $I_{I,q}$, $I_{\mathrm{FG},I,q}$, and $\omega_{I,q}$ are the components of the coarse-grained moment of inertia, fine-grained moment of inertia, and angular velocity about the $q$ axis, and $\langle \cdots \rangle_{\boldsymbol{R}_I,\boldsymbol{\Omega}_I}$ denotes an equilibrium average of fine-grained configurations consistent with the coordinate mapping of coarse-grained particle $I$. Furthermore, if the fluctuations in $I_{\mathrm{FG},I,q}$ are small compared to its mean, it can be shown that[21]

$$I_{I,q} \approx \left\langle I_{\mathrm{FG},I,q} \right\rangle_{\boldsymbol{R}_I,\boldsymbol{\Omega}_I}, \tag{3.10}$$

i.e. the principal moment of inertia of a coarse-grained particle about each principal axis $q$ is approximately equal to the equilibrium average of the corresponding principal moment of the fine-grained particles mapped onto it.

The AFM-CG method was derived only for the constant-volume conditions of the canonical ensemble, but is straightforwardly generalized to constant-pressure conditions by analogy with the MS-CG method for spherical coarse-grained particles in the isothermal-isobaric ensemble.[24] Thus, the force- and torque-matching conditions at constant pressure are the same as those in Eqs. (3.6) and (3.7), except that the coarse-grained forces, torques, and potential are in general functions of the coarse-grained system volume $V$ and the equilibrium average is constrained to configurations in which the fine-grained system volume $v = V$. The coarse-grained potential must also satisfy a virial-matching condition,[24]

$$\begin{aligned} W(\boldsymbol{R}^N, \boldsymbol{\Omega}^N, V) &= -\frac{\partial U}{\partial V} \\ &= \left\langle \frac{(n-N)k_{\mathrm{B}}T}{v} + \frac{1}{3v}\sum_{i=1}^{n} \boldsymbol{f}_i \cdot \boldsymbol{r}_i \right\rangle_{R^N,\Omega^N,V} \end{aligned} \tag{3.11}$$

In summary, for the equilibrium phase-space distribution of the coarse-grained model to match that of the fine-grained model in the isothermal-isobaric ensemble, the coarse-grained potential should satisfy Eqs. (3.6), (3.7), and (3.11), while the coarse-grained masses and principal moments of inertia should satisfy Eqs. (3.8) and (3.9), respectively. As shown below, using the more approximate Eq. (3.10) to parameterize the moments of inertia gives almost the same results as Eq. (3.9), even for a flexible molecule, so we have used this simpler equation for parameterization later on.

## 3.4 Methods

### 3.4.1 Force-, torque-, and virial-matching algorithm

The analytical expression for the coarse-grain potential $U$ is not usually known. However, an approximation to the functional form can be obtained using a neural-network optimization algorithm with Eqs. (3.6), (3.7), and (3.11) acting as necessary constraints. In general, $U(\boldsymbol{R}^N, \boldsymbol{\Omega}^N, V)$ is a function of the particle configuration and system volume. In this work, we have assumed that $U$ does not depend explicitly on $V$, in which case[24]

$$\frac{\partial U}{\partial V} = \frac{1}{3V}\sum_{I=1}^{N} \frac{\partial U}{\partial \boldsymbol{R}_I} \cdot \boldsymbol{R}_I. \tag{3.12}$$

With this approximation, the virial-matching condition in Eq. (3.11) can be written, using $v = V$, as

$$-\sum_{I=1}^{N} \frac{\partial U}{\partial \boldsymbol{R}_I} \cdot \boldsymbol{R}_I = \left\langle 3(n-N)k_{\mathrm{B}}T + \sum_{i=1}^{n} \boldsymbol{f}_i \cdot \boldsymbol{r}_i \right\rangle_{R^N,\Omega^N,V} . \tag{3.13}$$

Despite this approximation, we show that the coarse-grained models parameterized later on accurately match the average density of the corresponding all-atom fine-grained system at constant pressure.

To ensure that all equivalent configurations are assigned the same position in coordinate space, a transformation was made from the set of Cartesian coordinates to a vector $\boldsymbol{D}_{IJ}$ that was invariant under translation, rotation, and permutation of any pair of coarse-grained particles $I$ and $J$, [10,25–27] which was defined in terms of the positions, $\boldsymbol{R}_I$ and $\boldsymbol{R}_J$, and orientations, $\boldsymbol{\Omega}_I$ and $\boldsymbol{\Omega}_J$, of the two particles by

$$\begin{aligned}\boldsymbol{D}_{IJ} = \{&R_{IJ}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,1}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,2}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,3},\\ &\boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,3},\\ &\boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,3},\\ &\boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,3},\\ &\boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,3}\},\end{aligned} \tag{3.14}$$

where $R_{IJ} \equiv \|\boldsymbol{R}_{IJ}\|$, $\boldsymbol{R}_{IJ} \equiv \boldsymbol{R}_I - \boldsymbol{R}_J$ and $\boldsymbol{\Omega}_I$ and $\boldsymbol{\Omega}_J$ are specified by rotation matrices of the form of Eq. (3.2). The coordinates of each neighbor within the cut-off distance of particle $I$ were transformed to a $\boldsymbol{D}_{IJ}$ vector. All the $\boldsymbol{D}_{IJ}$ vectors for a given neighborhood were concatenated into a 2D matrix $\mathbb{D}_I$ of size $N \times \dim(\boldsymbol{D}_{IJ})$ representing a unique configurational fingerprint for coarse-grained particle $I$.

The potential function could then be written in terms of a set of neural network trainable parameters and activation functions transforming $\mathbb{D}_I$ to a potential energy value. While $\mathbb{D}_I$ is a sufficient specification of the coarse-grained coordinates to enforce relevant invariant properties of the molecular environment, it does not possess all the symmetries of the potential energy surface that it aims to fit. [25,28] For each molecular environment, it was assumed that the interactions were predominantly short-ranged such that neighbors beyond a certain cut-off distance, $R_{\mathrm{c}}$, do not contribute to the potential. [19] This condition can be enforced by a cut-off function of the form

$$g_{\mathrm{c}}(R_{IJ}) = \begin{cases} \frac{1}{2}\left[\cos\left(\frac{\pi R_{IJ}}{R_{\mathrm{c}}}\right) + 1\right], & R_{IJ} \leq R_{\mathrm{c}},\\ 0, & R_{IJ} > R_{\mathrm{c}}. \end{cases} \tag{3.15}$$

A set of these cut-off functions can enforce the radial symmetry conditions of the underlying potential energy surface by storing information about the radial distribution of neighbors according to [19]

$$G_I^1 = \sum_{J \neq I} g_{\mathrm{c}}(R_{IJ}). \tag{3.16}$$

Continuity of the potential along angular dimensions was ensured by using a compression layer to learn a set of collective variables from vector $\boldsymbol{D}_{IJ}$ which are constrained by the well-behaved modified $G^5$ symmetry function[19] given by

$$G_I^5 = \sum_{J \neq I} \prod_{\mu=1}^{M} 2^{1-\nu} \left(1 + \lambda \cos\theta_{IJ,\mu}\right)^{\nu} e^{-\eta(R_{IJ}-R_s)^2} g_{\text{c}}(R_{IJ}). \tag{3.17}$$

where $\lambda \in \{-1, 1\}$ and $R_{\text{s}}$, $\nu$, and $\eta$ are tunable hyperparameters and $\{\cos\theta_{IJ,\mu}\}$, is the set of machine-learned collective variables with the same properties as the angular component of the underlying potential and $M$ is the total number of machine-learned angular variables. These angular symmetry functions store information about the angular-radial distribution of neighbors in the local environment of coarse-grained particle $I$ Unlike the case of spherically symmetric particles, in a local reference frame, a neighboring anisotropic particle requires a minimum of seven independent scalar variables to fully describe its position and orientation. However, previous implementations of analytical potentials, including the Gay-Berne potential,[16,17] have used fewer coordinates for the calculation of the potential and forces. Similarly, for the neural network potential, an additional compression layer was included to remove the redundant angles from the $\boldsymbol{D}_{IJ}$ vectors, since the combination of translation and rotation in 3D is parameterized by at most 7 unique coordinates. The Behler symmetry functions were enforced on the output of the compression layer, ensuring that the learned compression had the same symmetry and continuity of the underlying potential. The reduction in the dimension of $\boldsymbol{D}_{IJ}$ also decreases the amount of data that is needed to train a sufficiently accurate potential. By removing the redundant angles in $\boldsymbol{D}_{IJ}$ there is a reduced possibility of over-fitting on a small data set.

A set of these symmetry functions with hyperparameters $(\lambda, \nu, \eta, R_{\text{s}}, R_{\text{c}})$ can be used to uniquely represent the structural fingerprint of the molecular environment. Symmetry functions used to represent the local environment were constructed using all possible permutations of values from a specified set of hyperparameters. Training of the neural network started with 8 symmetry functions and hyperparameters tuned to minimize the loss function, which is defined below. New symmetry functions were added to the set if they resulted in a significant reduction in the neural-network loss compared with the preceding iteration. The set of hyperparameters in the symmetry functions used in the anisotropic coarse-grained models parameterized in this work can be found in the Supplementary Material.

To further reduce the amount of data needed to train the neural network, a prior repulsive potential was defined with pairwise additive properties. This potential was used to ensure physical behavior in regions of the potential where the forces are large and thus are rarely sampled in an equilibrium molecular dynamics simulation. This prior potential only needs to satisfy two conditions: firstly, it must be repulsive at short radial separations, and, secondly, the position of its repulsive barrier must be orientationally dependent. A simple equation satisfying these conditions is

$$U_{\text{prior},I} = \sum_{J \neq I} B_1 \sigma_{\text{c}} \left(\mathbb{D}_I\right)^{-B_2}, \tag{3.18}$$

where $\sigma_{\text{c}}$ is a neural-network compression layer function and $B_1$ and $B_2$ are strictly positive trainable parameters. It is also possible to achieve a similar large repulsive barrier through a more advanced non-linear sampling of the molecular dynamics simulation data. $U_{\text{prior}}$ fits a purely repulsive potential with angular dependence to the molecular environment, while $U_{\text{NN}}$ fits the attractive and oscillatory corrections to the environment. The final prediction for the potential energy of the environment of coarse-grained particle $I$ is therefore the sum of the neural network potential and the prior repulsive potential,[15]

$$U_I = U_{\mathrm{NN},I} + U_{\mathrm{prior},I}, \tag{3.19}$$

and, thus, the total coarse-grained potential is

$$U = \sum_{I=1}^{N} U_I \tag{3.20}$$

From the matching conditions in Eqs. (3.6), (3.7), and (3.13), optimization of the neural-network weights and biases requires a loss function of the form

$$\begin{aligned} L = \Bigg\langle \sum_{I=1}^{N} \Bigg( \alpha \left| \boldsymbol{F}_{\mathrm{FG},I} + \frac{\partial U}{\partial R_I} \right|^2 + \beta \left| \boldsymbol{\tau}_{\mathrm{FG},I} + \sum_q \Omega_{I,q} \times \frac{\partial U}{\partial \Omega_{I,q}} \right|^2 \Bigg) \\ + \gamma \left| 3\,(n-N)\,k_{\mathrm{B}}T + \sum_{I=1}^{N} \left( \bar{W}_{\mathrm{FG},I} + \frac{\partial U}{\partial \boldsymbol{R}_I} \cdot \boldsymbol{R}_I \right) \right|^2 \Bigg\rangle_{R^N,\Omega^N,V}, \end{aligned} \tag{3.21}$$

where

$$\boldsymbol{F}_{\mathrm{FG},I} \equiv \sum_{i \in \zeta_I} \boldsymbol{f}_i,\ \boldsymbol{\tau}_{\mathrm{FG},I} \equiv \sum_{i \in \zeta_I} \Delta \boldsymbol{r}_i \times \boldsymbol{f}_i,\ \bar{W}_{\mathrm{FG},I} \equiv \sum_{i \in \zeta_I} \boldsymbol{f}_i \cdot \boldsymbol{r}_i, \tag{3.22}$$

and $\alpha, \beta$, and $\gamma$ are weights. These weights specify the fraction of each loss that is used for backpropagation and were free to change with the learning rate during optimization.[14] Even though there have been significant efforts in the development of methods to fit the averaged coarse-grained forces directly,[29,30] the average total fine-grained forces subject to the constraint of matching fine-grained and coarse-grained configurations are not easily obtained. An indirect means of minimizing the loss function in Eq. (3.21) above is possible by replacing the constrained ensemble average with an average over instantaneous unconstrained simulation configurations,[14]

$$\begin{aligned} L_{\mathrm{inst}} = \sum_{t=1}^{N_{\mathrm{t}}} \Bigg[ \sum_{I=1}^{N} \Bigg( \alpha \left| \boldsymbol{F}_{\mathrm{FG},I}(\boldsymbol{r}_t^n) + \frac{\partial U(\boldsymbol{\xi}_t)}{\partial R_I} \right|^2 \\ + \beta \left| \boldsymbol{\tau}_{\mathrm{FG},I}(\boldsymbol{r}_t^n) + \sum_q \Omega_{I,q}(\boldsymbol{\xi}_t)) \times \frac{\partial U(\boldsymbol{\xi}_t))}{\partial \Omega_{I,q}} \right|^2 \Bigg) \\ + \gamma \left| 3\,(n-N)\,k_{\mathrm{B}}T + \sum_{I=1}^{N} \left( \bar{W}_{\mathrm{FG},I}(\boldsymbol{r}_t^n) + \frac{\partial U(\boldsymbol{\xi}_t)}{\partial \boldsymbol{R}_I} \cdot \boldsymbol{R}_I(\boldsymbol{\xi}_t) \right) \right|^2 \Bigg], \end{aligned} \tag{3.23}$$

since it can be shown, for a sufficiently large dataset that comprehensively samples the equilibrium ensemble of the fine-grained system, that $L$ and $L_{\mathrm{inst}}$ have the same global minimum. Here, $N_{\mathrm{t}}$ is the number of simulation configurations in the dataset, $\boldsymbol{r}_t^n$ and $v_t$ are the fine-grained coordinates and system volume for configuration $t$, and $\boldsymbol{\xi}_t = (\boldsymbol{R}^N(\boldsymbol{r}_t^n), \boldsymbol{\Omega}^N(\boldsymbol{r}_t^n), V(v_t))$ is the mapped coarse-grained configuration for this fine-grained configuration. The loss function is optimized using the minibatch gradient descent as implemented in TensorFlow.

The feedforward neural network shown in Fig. 3.1 was then trained, where the forward propagation used matrix $\mathbb{D}_I$ as an input to predict the coarse-grained potential $U$, after which TensorFlow's computational derivative was used to calculate the outputs, namely the predicted forces, torques, and virial. In the backpropagation stage, the loss function was used to calculate the error between the true and predicted values, which was then used to update the network weights and biases. The errors between the true and predicted parameters were calculated using TensorFlow's mean squared error, and gradient descent was implemented using TensorFlow's Adam optimizer.[31] Once the error of the neural network was minimized, the neural network model was used to predict the forces, torques, and virial. However, removing the output and derivative layers gives access to the predicted potential of mean force. By optimizing the partial derivatives of the potential instead of the potential itself, by the nature of the derivative, there will be less oscillation in the potential at the edges of the data set close to the cut-off distances.

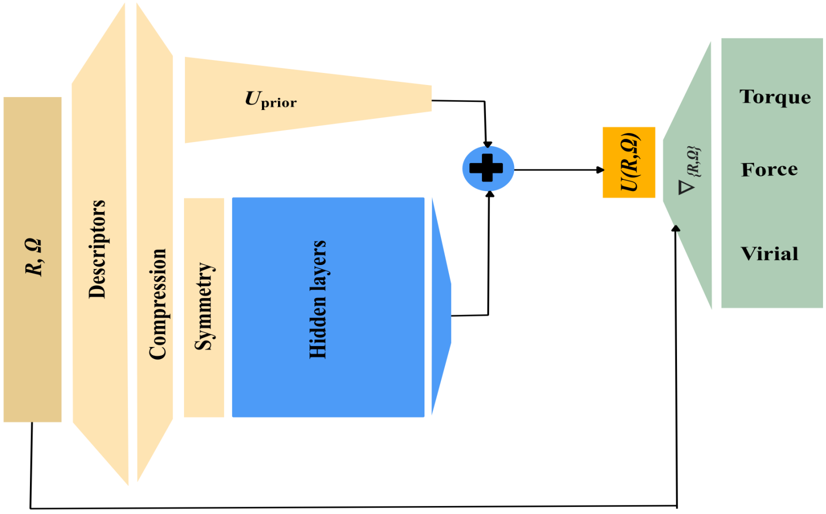


**Figure 3.1:** Schematic of anisotropic force-matching neural network architecture.

### 3.4.2 LAMMPS modification and neural network implementation

The neural network was constructed in Tensorflow (version 2.3.0)[32] using the Keras (version 2.4.3) functional API[33] and saved using the Tensorflow SavedModel format. The trained neural network was implemented in LAMMPS using the Tensorflow C API and cppflow wrapper. All simulations were carried out using the LAMMPS molecular dynamics (MD) software package (version 20Nov19).[34–36] The Optimized Potentials for Liquid Simulations-All Atom (OPLS-AA) force field[37–40] was used for all all-atom simulations with a cut-off distance of 10 Å for short-ranged non-bonded interactions; long-ranged electrostatic interactions were calculated with the particle-particle particle-mesh (PPPM) method[36,41] The bonds that include hydrogen were constrained using the SHAKE algorithm.[42] Simulations were carried out in the isothermal-isobaric (NPT) ensemble at a pressure of 1 atm, with the temperature and pressure controlled by a Nosé-Hoover thermostat and barostat.[43,44]

Neural network training was carried out using data from a 25 ns all-atom simulation in which simulation configurations and forces and velocities were saved at 2 ps intervals. The simulation snapshots from the last 20 ns were shuffled and then divided into 4 groups of equal size, $\{g_0, g_1, g_2, g_3\}$. The neural network was initially trained on $g_0$ and validated on $g_3$. The validation set $g_3$ was further divided into an 8:2 ratio where the lesser was reserved as the test set. New snapshots were added from $g_1$ and $g_2$ if the mean errors of their predicted forces and torques were larger than that of the test set. The accuracy of the trained neural network was then compared to the expected accuracy determined from k-fold cross-validation.[45,46] During k-fold cross-validation, the last 20 ns of simulation data was shuffled and divided into 10 folds, $\{\psi_0, ..., \psi_9\}$. The model was validated on $\psi_i$

and trained on $\bigcup_{j\neq i}\psi_j$ for all $i,j\in\{0-9\}$. The loss of the iterative training method was found to be identical to the k-fold cross-validation loss.

The coarse-grained simulations were done using a modified version of the LAMMPS software where the trained neural network was introduced to calculate the forces and energies. The dimensions of the coarse-grained sites used in the simulations were derived from the inertia tensor of the all-atom model. To test the ability of the coarse-grained model to capture the properties of the all-atom model under a variety of conditions in addition to the single temperature at which the neural network was trained, the equilibrium structural properties of equivalent coarse-grained and all-atom systems were compared in simulations at several different temperatures. In all cases, the total length of the coarse-grained simulation was 25 ns long with the last 20 ns being used to calculate all structural properties. The timestep of all coarse-grained simulations was also set to 12 fs.

## 3.5 Results and Discussion

To demonstrate the flexibility of the method we have used our neural-network model to construct coarse-grained interaction potentials for benzene, an archetypal anisotropic small molecule, and $\alpha$-sexithiophene, an organic semiconductor with significant applications in organic electronic devices[47–49] (Fig. 3.2). These molecules were selected to demonstrate the neural network's ability to handle anisotropic molecules of varying complexity, flexibility, and aspect ratio while still reproducing the structural and phase behavior.

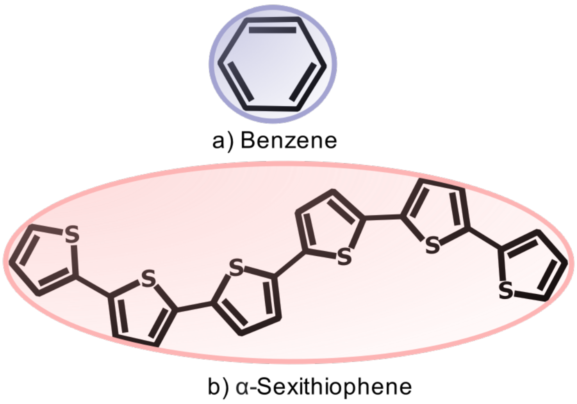


**Figure 3.2:** Chemical structures of (a) benzene and (b) $\alpha$-sexithiophene with coarse-grained ellipsoid superimposed on one possible configuration of each molecule.

The shape of a coarse-grained particle obtained from the AFM-CG method is determined by the "average" shape of the fine-grained molecule or molecular fragment that is mapped to it under the parameterization conditions. Thus, the variation of the aspect ratio of the molecule or molecular fragment with temperature in the all-atom simulations can potentially be used as a qualitative indicator of the temperature transferability of the coarse-grained model. Here, the aspect ratio of the molecule was calculated as the ratio of the length to the breadth of the molecule, where the length was defined as the longest principal axis and the breadth was defined as the sum of the remaining two semi-axes. Unlike benzene, the thiophene-thiophene torsion angles also have a temperature-dependent effect on the aspect ratio of sexithiophene.

Neural networks in general are very good at interpolation but struggle with extrapolation[50–53]. The accuracy of the model is therefore expected to decrease as the aspect ratio of the molecule deviates from that at the parameterization temperature, as well as when the density distribution is sufficiently different from the parameterization temperature. By parameterizing the systems in the liquid phase, the model can capture a wider variety of fluctuations in the density of the system and the dimensions of the molecules. The average size of a flexible molecule in the isotropic phase will be different from the size of the molecule when locked in a rigid crystal structure.[54,55] However, this temperature-dependent size difference should decrease with increased rigidity of the molecule.

### 3.5.1 Benzene

Simulations consisting of 500 benzene molecules were carried out at 280, 300, 320, 330, and 350 K, and the coarse-grained neural-network model was parameterized at 300 K. The time step was 2 fs in the all-atom simulations and 12 fs in the coarse-grained simulations. The cut-off distance hyperparameter $R_c$ was 10 Å. The root mean squared validation error for the forces was 2.55 kcal mol$^{-1}$ Å$^{-1}$ and that of the torque was 4.35 kcal mol$^{-1}$. The average post-training error in the pressure calculated for the entire simulation box volume and over the entire length of the simulation at the parameterization temperature was 0.0092 atm. Benzene's average principal moments of inertia in the all-atom simulation at 300 K were used to determine the principal moments of the coarse-grained benzene model using Eqn. (3.10) (values given in the Supplementary Material) since fluctuations in the moments at the parameterization temperature were small.[21]

The variation of the molecular aspect ratio of the all-atom benzene model with temperature is shown in Fig. 3.3. The distribution of possible dimensions observed for benzene is narrow and remains fairly constant with temperature, making benzene an ideal case where molecular flexibility does not contribute significantly to the overall error of the model.[56]

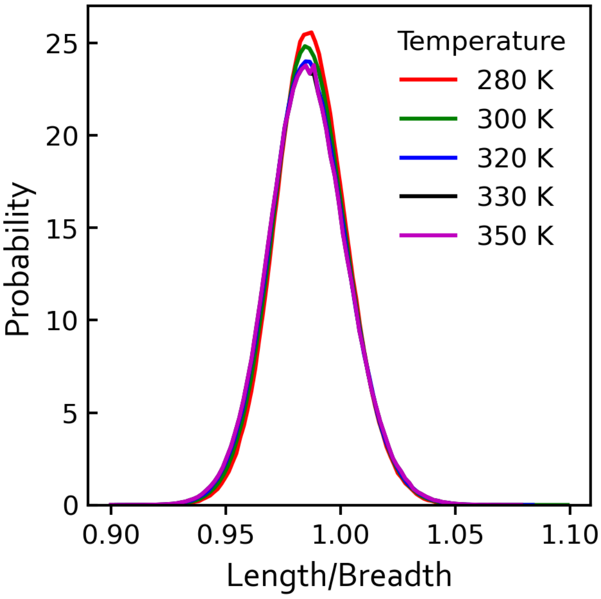


**Figure 3.3:** Length-to-breadth ratio of the all-atom benzene model at 1 atm and various temperatures.

Fig. 3.4 shows that the coarse-grained neural-network model accurately captures the liquid density of the all-atom model over a wide range of temperatures from just above the freezing point to just below the boiling point, with only slight deviations for the temperature furthest from the parameterization temperature. As shown in Fig. 3.5, the coarse-grained model also accurately predicts the radial distribution function (RDF) of the all-atom model over the same temperature range.

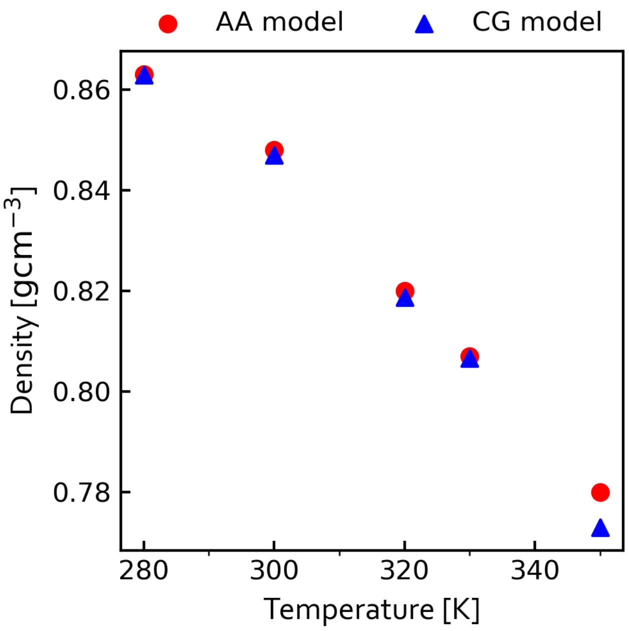


**Figure 3.4:** Density versus temperature of the all-atom (AA) and coarse-grained (CG) benzene models at 1 atm. Error bars are smaller than the symbol

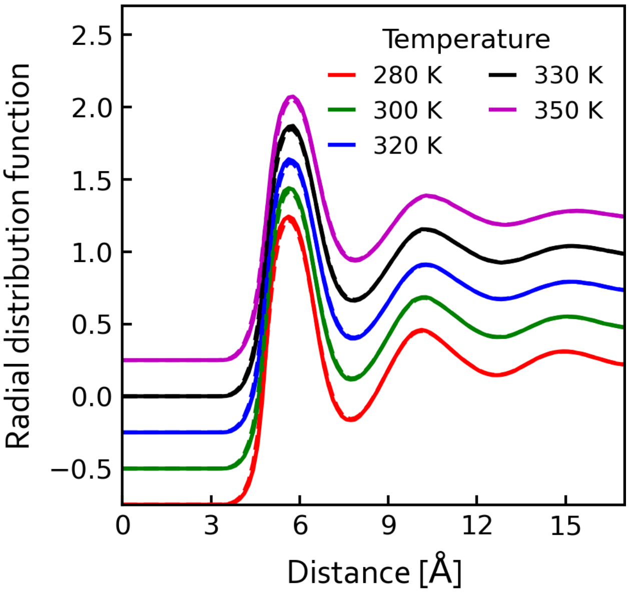


**Figure 3.5:** Radial distribution function (RDF) of the all-atom (solid lines) and coarse-grained (dashed lines) benzene models at 1 atm and various temperatures. The RDFs have been shifted vertically for clarity.

To further elucidate the accuracy of the neural network coarse-grained model, the angular-radial distribution function (ARDF) was analyzed. The ARDF is defined by

$$g(r,\theta) = \frac{\langle n(r,\theta)\rangle}{\frac{4}{3}\pi\rho[(r+\Delta r)^3 - r^3]\sin\theta\Delta\theta}, \tag{3.24}$$

where $\langle n(r,\theta)\rangle$ is the average number of molecules in the spherical shell within the bounds $r$ to $r+\Delta r$ of the center-of-mass of a chosen molecule and having an out-of-plane axis rotation of $\theta$ with respect to the out-of-plane axis of the chosen molecule[57], and $\rho$ is the bulk number density. Fig. 3.6 shows the 2D heatmap of the ARDF along with 1D slices of this function at specific angles at 300 K (the parameterization temperature) for both the all-atom and coarse-grained models. The ARDFs at the other simulated temperatures are compared in the Supplementary Material. At all simulated temperatures between 280 and 350 K, the coarse-grained model captures all the major features of the fine-grain structure of the fluid. The only difference is a slight underestimation of the peak heights by the coarse-grained model. The neural-network model is, however, able to more faithfully capture the angular radial distribution of benzene at all temperatures compared with the coarse-grained benzene model previously parameterized with the AFM-CG method using a pair potential to describe the interactions between coarse-grained particles.[21] This improvement can be attributed to the greater flexibility of the neural-network potential in describing the intermolecular interactions. The neural-network model can demonstrate improved temperature transferability by adjusting the neural network hyperparameters to prevent overfitting of the local number density variations.

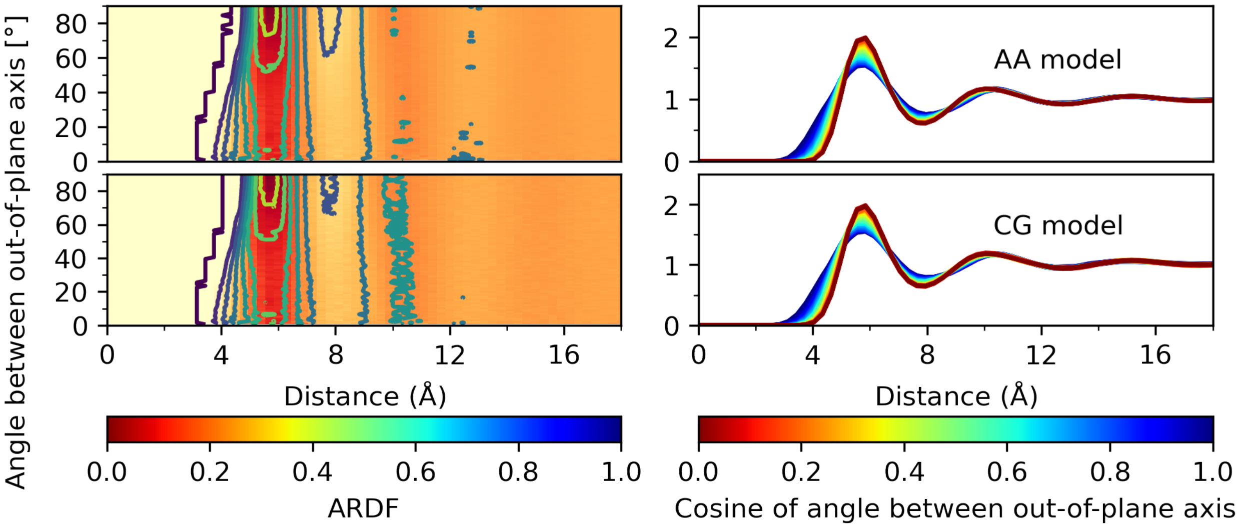


**Figure 3.6:** Angular-radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) benzene models at 300 K and 1 atm depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on, or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90°.

The coarse-grained simulation of anisotropic molecules using a neural network potential is more suited for large, preferably rigid, molecules, for which a high degree of coarse-graining can be achieved. However, the model was still able to achieve a modest 20× speedup, through a combination of reduced computation time per timestep and a larger timestep, when compared to the atomistic simulations. This poor performance for a small molecule such as benzene is due to the small reduction in the number of degrees of freedom from the all-atom model to the coarse-grained model, coupled with a neural-network potential that is more computationally expensive than an analytical potential. Nevertheless, computational savings are obtained even in this suboptimal case. Simulations were carried out on a 4-core Intel i7-4790K CPU, but, further speedups could be achieved by taking advantage of the GPU-enabled version of TensorFlow.

### 3.5.2 Sexithiophene

Simulations of 512 sexithiophene molecules were carried out at 570, 590, 640, and 680 K temperatures, corresponding to temperatures previously identified in all-atom MD simulations to correspond to crystalline (K), smectic-A (Sm-A), nematic (N), and isotropic (I) phases respectively.[58] The time step was 1 fs in the all-atom simulations and 12 fs in the CG simulations. Although we have used the OPLS-AA force field for our all-atom simulations, whereas these previous MD simulations[58] used the related AMBER force field[59–61] the structural properties of systems simulated with these two force fields (in particular the density, orientational order parameter, and radial distribution function discussed below) are very similar for the temperature range studied. The cut-off distance hyperparameter $R_\mathrm{c}$ was set to 21 Å. The neural network model was parameterized using simulation snapshots from the isotropic phase at 680 K, where the molecular mobility was highest. The conditions of the isotropic bulk phase are advantageous in efficiently sampling the configuration space, especially rare high-energy configurations necessary for the accurate reproduction of the repulsive part of the coarse-grained potential. As shown in Fig. 3.7a, the distributions of the principal moments of inertia of sexithiophene in the all-atom simulation at the parameterization temperature are broad, indicating that Eqn. (3.10). may not be adequate for parameterizing the moments of inertia of the coarse-grained model. However, we found that using the more general Eqn. (3.9) to parameterize the coarse-grained moments of inertia (by fitting the distributions in Fig. 3.7(b–d) gave values within <1%. So we used the values from Eqn. (3.9) in the coarse-grained model."

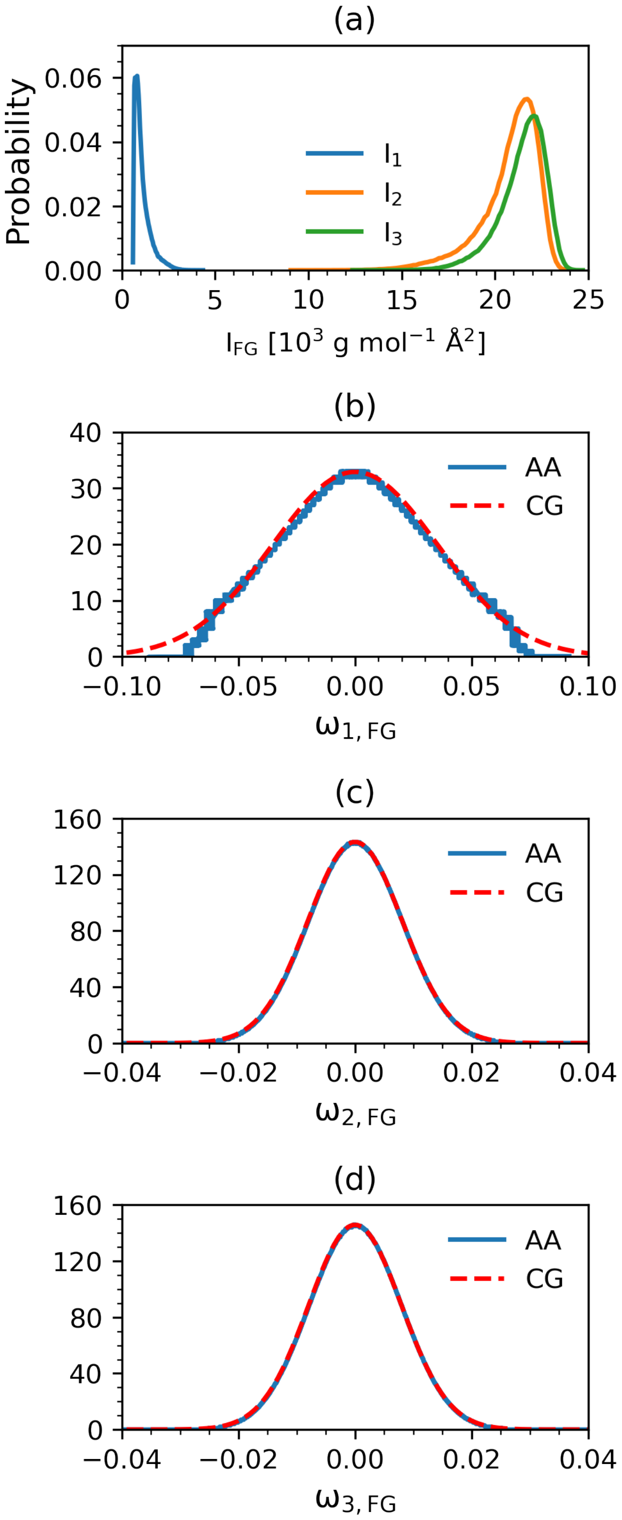


**Figure 3.7:** (a) Principal moment of inertia distributions for all-atom (AA) sexithiophene model at 680 K and 1 atm. The corresponding angular velocity distributions of each principal axis along with the coarse-grained (CG) fit to the distribution given by Eq. (3.9) is shown in (b)–(d).

The root mean squared validation error for the sexithiophene forces were 3.95 kcal mol$^{-1}$ Å$^{-1}$ and that of the torque was 9.8 kcal mol$^{-1}$. The sexithiophene final force and torque losses were larger than those of benzene because the model was not complex enough to account for the bending of the polymer and the rotation of the individual thiophene rings. The loss is also skewed to larger values when compared with benzene because sexithiophene is a larger molecule and so the interactions between molecules are stronger overall.

The structural properties of the coarse-grained model were compared to those of its all-atom counterpart at each of the simulated temperatures. The nonlinear change in density with respect to temperature is associated with the phase changes that occur at the simulated temperatures (Fig. 3.8).[58] The density of the coarse-grained system agrees well with that of the all-atom system, with minimal deviations from the fine-grained system with increasing distance from the parameterization temperature. Compared with benzene, sexithiophene has a much larger change in density between the crystalline and the isotropic phase. This difference results in less overlap between the local density variations in the crystalline phase at the lowest temperature and the training data set in the isotropic phase at the highest temperature. The sexithiophene molecule is also much more flexible than benzene, as seen in the wide distribution of the aspect ratio in the all-atom model at all the simulated temperatures shown in Fig. 3.9, and its dimensions change significantly with temperature over the range studied. Another limitation of representing sexithiophene as a single-site ellipsoid is the loss of thiophene–thiophene torsional information. That is, for any given position and orientation of the coarse-grained ellipsoid there are multiple different relative orientations between the thiophene groups.[62] This loss of information is significant because the anisotropic interactions of the thiophene subunits are lost, which reduces the neural network's ability to isolate which of the two short axes corresponds to the $\pi$-stacking direction.

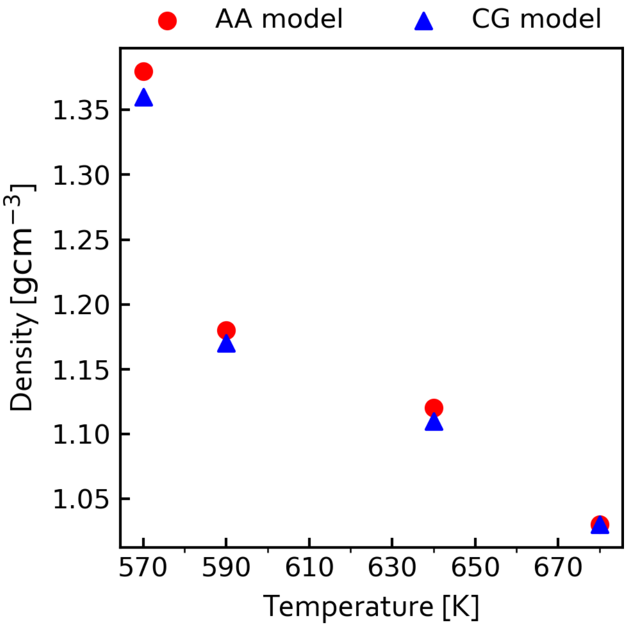


**Figure 3.8:** Density versus temperature of the all-atom (AA) and coarse-grained (CG) sexithiophene models at 1 atm. Error bars are smaller than the symbols.

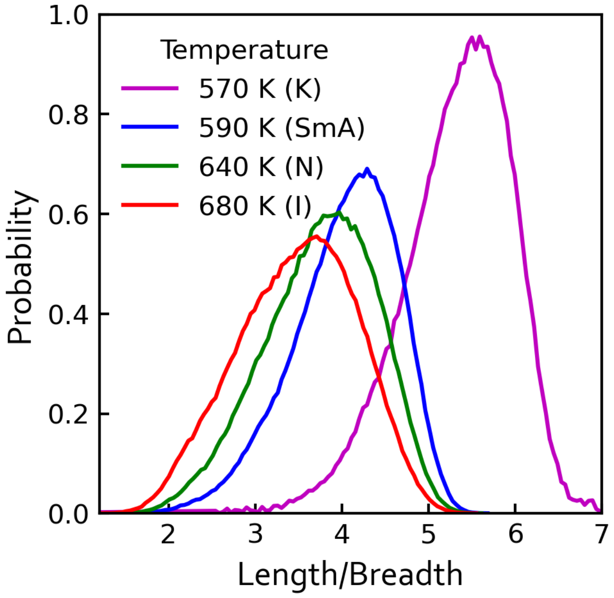


**Figure 3.9:** Length-to-breadth ratio of all-atom sexithiophene model at 1 atm and various temperatures. The simulated phase is given in parentheses after each temperature in the legend (I = isotropic, N = nematic, SmA = smectic A, K = crystal).

To further confirm that the density changes were associated with transitions from crystalline through nematic and smectic to the isotropic phase, the scalar orientational order parameter $P_2$ was introduced. For a given simulation snapshot at time $t$, $P_2$ can be found by diagonalizing the ordering matrix $\boldsymbol{Q}$,

$$\boldsymbol{Q}(t) = \frac{1}{2N}\sum_{I=1}^{N}[3\boldsymbol{u}_I(t)\otimes\boldsymbol{u}_I(t) - \boldsymbol{E}], \tag{3.25}$$

where $\boldsymbol{u}_I$ is the unit vector along the molecular axis and $\boldsymbol{E}$ is the identity matrix. $\langle P_2 \rangle$ is the average over the largest eigenvalue of this matrix for all snapshots of equilibrium configurations.[58] Larger values of the scalar orientational order parameter close to one indicate an ordered crystalline structure while values close to zero correspond to an isotropic disordered phase. The coarse-grained model reproduces the orientational order parameter of the all-atom model reasonably well over the temperature range simulation, as shown in Fig. 3.10. The coarse-grained model underestimates the degree of orientational ordering observed in the all-atom model away from the parameterization temperature, likely because it does not capture the increasing molecular shape anisotropy that is observed in the all-atom model as the temperature decreases (Fig. 3.9). As expected, the largest difference occurs in the predicted crystalline phase.

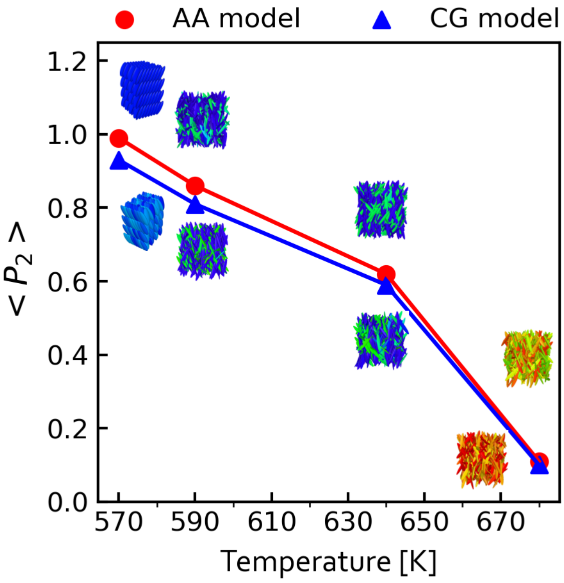


**Figure 3.10:** Orientational order parameter versus temperature for the all-atom (AA) and coarse-grained (CG) sexithiophene models at 1 atm. Typical simulation configurations are shown at each temperature for each system (AA model above the data points and CG model below), in which the molecules have been colored according to their orientation with respect to the phase director (blue = parallel, red = perpendicular). Error bars are smaller than the symbols.

The same trend is seen in the radial distribution functions shown in Fig. 3.11, in which the agreement between the coarse-grained and all-atom models at most temperatures is excellent, with the largest deviations for the crystalline phase. The underestimation and broadening of the peaks in the crystalline radial distribution function explain the discrepancy between the order parameter of the all-atom and coarse-grained models. The observed differences are most likely due to the effect on molecular packing of the aforementioned discrepancy in molecular shape between the two models as temperature decreases.[63] Nevertheless, even in the crystalline phase, the coarse-grained model captures the peak positions of the radial distribution function very well.

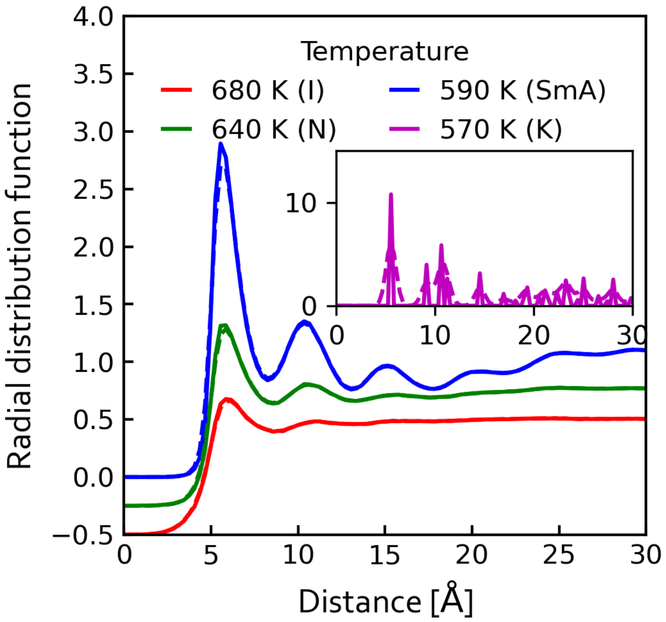


**Figure 3.11:** Radial distribution function (RDF) of the all-atom (solid lines) and coarse-grained (dashed lines) sexithiophene models at 1 atm and various temperatures. The RDFs have been shifted vertically for clarity. The simulated phase is given in parentheses after each temperature in the legend (I = isotropic, N = nematic, SmA = smectic A, K = crystal).

The coarse-grained model also accurately describes orientational correlations in condensed-phase sexithiophene, as illustrated by a comparison with the angular-radial distribution function of the all-atom model. At the parameterization temperature, the coarse-grained model is able to capture all major features when compared to the all-atom model (Fig. 3.12). The neural-network model is also able to capture the relevant features in the structure of sexithiophene's smectic liquid-crystal phase at 590 K, as shown in Fig. 3.13. The discrepancies in the width and height of the peaks are likely due to the differences in molecular shape away from the parameterization temperature that was mentioned earlier. The ARDFs of the two models at 640 K are compared in the Supplementary Material and show similarly good agreement.

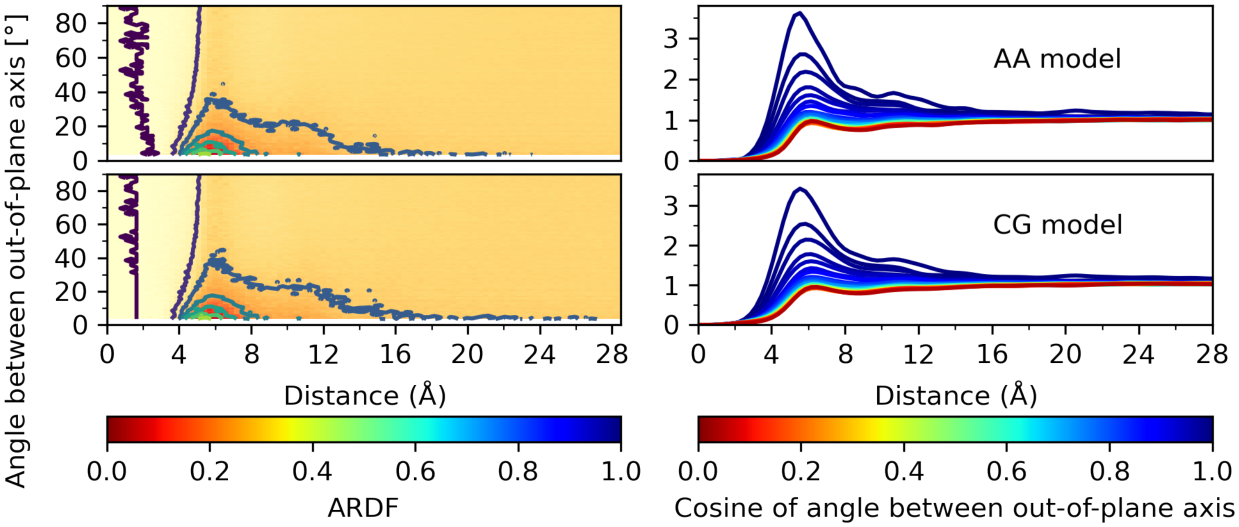


**Figure 3.12:** Angular-radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) sexithiophene models at 680 K and 1 atm (isotropic phase) depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on, or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90 °.

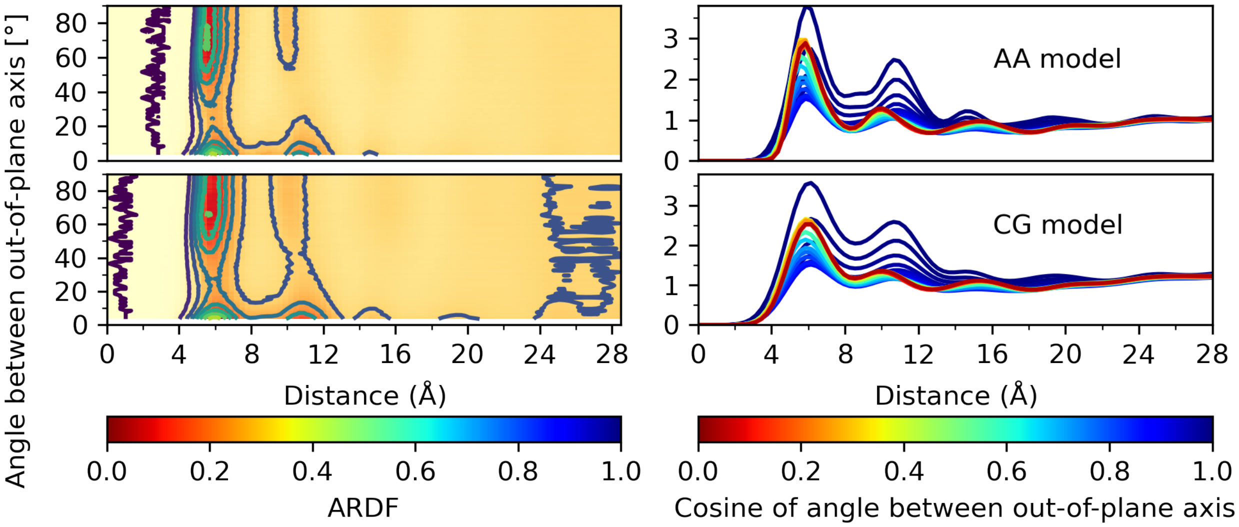


**Figure 3.13:** Angular-radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) sexithiophene models at 590 K and 1 atm (smectic phase) depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on, or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90 °.

Despite sexithiophene not strictly meeting the conditions to be coarse-grained to a single anisotropic particle due to its significant flexibility, the coarse-grained neural-network model is still able to reproduce its condensed-phase structural properties and phase behavior with remarkable accuracy. The limitation of the single-site model is only evident under conditions where the conformation of the molecule is highly temperature-dependent. One way to construct a neural network model that is independent of temperature would be to extract the training data from multiple temperatures and define the molecular dimensions as the average over the crystalline and isotropic phases. While the results for sexithiophene are substantially better than expected given its flexibility, improvements can be made to the model by considering a coarse-grained mapping consisting of more than one site.[64]

The coarse-grained simulation of sexithiophene demonstrated a speed-up of 132× compared to the all-atom simulation using the same hardware employed for the benzene simulations. This speedup is primarily due to the large reduction in the number of degrees of freedom in coarse-graining this molecule.

## Conclusions

We have applied machine learning and a recently derived systematic coarse-graining method for anisotropic particles to develop a single-site anisotropic coarse-grained potential of a molecular system. The iterative training of the neural network potential is able to reproduce the forces, torques, and pressure of the fine-grained all-atom system. The final loss of the iterative training model was identical to the loss obtained from k-fold cross-validation. The CG model performs well for a rigid molecule like benzene but remarkably it also describes the phase behavior and molecular-scale structural correlations of a flexible molecule like sexithiophene with comparable accuracy, even though the aspect ratio of the molecule changes significantly over the simulated temperature range. We have demonstrated the versatility of the coarse-graining method by parameterizing models of benzene and sexithiophene at a single temperature and then studying their accuracy in capturing the structural properties of the corresponding all-atom model at different temperatures. The sexithiophene model was also used to show the ability of the model to reproduce the phase behavior of the all-atom model, with the lowest fidelity coming from the crystalline phase where the aspect ratio of the molecule has the largest deviation from the parameterization data set. A natural extension to this work would be to generalize the method to a multi-site anisotropic coarse-grained model for flexible molecules and polymers.

# References

[1] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Nat. **559**, 547 (2018).

[2] S. M. Moosavi, K. M. Jablonka, and B. Smit, J. Am. Chem. Soc. **142**, 20273 (2020).

[3] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, and S. Velankar, Nucl. Acids Res. **50**, D439 (2021).

[4] P. Rowe, V. L. Deringer, P. Gasparotto, G. Csányi, and A. Michaelides, J. Chem. Phys. **153**, 034702 (2020).

[5] S. Stocker, J. Gasteiger, F. Becker, S. Günnemann, and J. T. Margraf, Mach. Learn.: Sci. Tech. **3**, 045010 (2022).

[6] P. Friederich, F. Häse, J. Proppe, and A. Aspuru-Guzik, Nat. Mater. **20**, 750 (2021).

[7] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, Ann. Rev. Phys. Chem. **71**, 361 (2020).

[8] Z. Guo, D. Lu, Y. Yan, S. Hu, R. Liu, G. Tan, N. Sun, W. Jiang, L. Liu, Y. Chen, L. Zhang, M. Chen, H. Wang, and W. Jia, in *Proc. 27th ACM SIGPLAN Symp. Princ. Pract. Parallel Program.*, PPoPP '22 (Association for Computing Machinery, New York, NY, USA, 2022) pp. 205–218.

[9] D. Marx and J. Hutter, *Ab initio molecular dynamics: basic theory and advanced methods* (Cambridge University Press, 2009).

[10] H. Wang, L. Zhang, J. Han, and W. E, Comput. Phys. Commun. **228**, 178 (2018).

[11] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, Chem. Rev. **121**, 10142 (2021).

[12] J. Jin, A. J. Pak, A. E. Durumeric, T. D. Loose, and G. A. Voth, J. Chem. Theory Comput. **18**, 5759 (2022).

[13] S. J. Marrink and D. P. Tieleman, Chem. Soc. Rev. **42**, 6801 (2013).

[14] L. Zhang, J. Han, H. Wang, R. Car, and W. E. Weinan, J. Chem. Phys. **149**, 034101 (2018).

[15] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. D. Fabritiis, F. Noé, and C. Clementi, ACS Cent. Sci. **5**, 755 (2019).

[16] R. Berardi, C. Fava, and C. Zannoni, Chem. Phys. Lett. **236**, 462 (1995).

[17] J. G. Gay and B. J. Berne, J. Chem. Phys. **74**, 3316 (1981).

[18] B. J. Boehm, H. T. Nguyen, and D. M. Huang, J. Phys.: Cond. Matter **31**, 423001 (2019).

[19] J. Behler, J. Chem. Phys. **134**, 074106 (2011).

[20] G. Campos-Villalobos, G. Giunta, S. Marín-Aguilar, and M. Dijkstra, J. Chem. Phys. **157**, 024902 (2022).

[21] H. T. L. Nguyen and D. M. Huang, J. Chem. Phys. **156**, 184118 (2022).

[22] W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, J. Chem. Phys. **128**, 244114 (2008).

[23] H. Goldstein, *Classical Mechanics* (Addison-Wesley San Francisco, 2002).

[24] A. Das and H. C. Andersen, J. chem phys **132**, 164106 (2010).

[25] A. P. Bartók, R. Kondor, and G. Csányi, Phys. Rev. B **87**, 184115 (2013).

[26] J. Han, L. Zhang, R. Car, and W. E, Commun. Comput. Phys. **23**, 629 (2018).

[27] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, in *2019 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR* (IEEE, 2019) pp. 5738–5746.

[28] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, J. Chem. Phys. **148**, 241709 (2018).

[29] G. Ciccotti, R. Kapral, and E. Vanden-Eijnden, ChemPhysChem **6**, 1809 (2005).

[30] J. B. Abrams and M. E. Tuckerman, J. Phys. Chem. B **112**, 15742 (2008).

[31] D. P. Kingma and J. Ba, arXiv , 1412.6980 (2014).

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, in *12th USENIX Symp. Oper. Syst. Des. Implement. OSDI 16* (2016) pp. 265–283.

[33] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras (2015).

[34] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[35] W. M. Brown, P. Wang, S. J. Plimpton, and A. N. Tharrington, Comput. Phys. Commun. **182**, 898 (2011).

[36] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington, Comput. Phys. Commun. **183**, 449 (2012).

[37] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

[38] W. L. Jorgensen and N. A. McDonald, J. Mol. Struct. THEOCHEM **424**, 145 (1998).

[39] R. C. Rizzo and W. L. Jorgensen, J. Am. Chem. Soc. **121**, 4827 (1999).

[40] M. L. Price, D. Ostrovsky, and W. L. Jorgensen, J. Comput. Chem. **22**, 1340 (2001).

[41] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (CRC Press, 1998).

[42] J.-P. Ryckaert, G. Ciccotti, and H. J. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[43] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[44] S. Nosé, Mol. Phys. **52**, 255 (1984).

[45] B. G. Marcot and A. M. Hanea, Comput. Statist. **36**, 2009 (2021).

[46] Y. Bengio and Y. Grandvalet, Adv. Neural Inf. Process. Syst. **16** (2003).

[47] H. Katz, J. Mater. Chem. **7**, 369 (1997).

[48] D. Fichou, J. Mater. Chem. **10**, 571 (2000).

[49] Y. Dong, V. C. Nikolis, F. Talnack, Y.-C. Chin, J. Benduhn, G. Londi, J. Kublitski, X. Zheng, S. C. Mannsfeld, D. Spoltore, *et al.*, Nat. commun. **11**, 1 (2020).

[50] P. J. Haley and D. Soloway, in *Proc. 1992 IJCNN Int. Jt. Conf. Neural Netw.*, Vol. 4 (IEEE, 1992) pp. 25–30.

[51] G. S. Na, S. Jang, and H. Chang, Phys. Chem. Chem. Phys. **24**, 1300 (2022).

[52] Y. Ding, A. Pervaiz, M. Carbin, and H. Hoffmann, in *Proc. 29th ACM Jt. Meet. Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.* (2021) pp. 728–740.

[53] J. P. Rigol, C. H. Jarvis, and N. Stuart, Int. J. Geogr. Inf. Sci. **15**, 323 (2001).

[54] M. Mueller, J. Zierenberg, M. Marenz, P. Schierz, and W. Janke, Phys. Procedia **68**, 95 (2015).

[55] D. Seaton, S. Mitchell, and D. Landau, Braz. J. Phys. **36**, 623 (2006).

[56] A. V. Sinitskiy, M. G. Saunders, and G. A. Voth, J. Phys. Chem. B **116**, 8363 (2012).

[57] M. Falkowska, D. T. Bowron, H. G. Manyar, C. Hardacre, and T. G. Youngs, ChemPhysChem **17**, 2043 (2016).

[58] A. Pizzirusso, M. Savini, L. Muccioli, and C. Zannoni, J. Mater. Chem. **21**, 125 (2011).

[59] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner, J. Am. Chem. Soc. **106**, 765 (1984).

[60] S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, J. Comput. Chem. **7**, 230 (1986).

[61] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, J. Am. Chem. Soc. **117**, 5179 (1995).

[62] F. D. Tsourtou, S. D. Peroukidis, L. D. Peristeras, and V. G. Mavrantzas, Macromol. **51**, 8406 (2018).

[63] W. Xia, N. K. Hansoge, W.-S. Xu, F. R. Phelan Jr, S. Keten, and J. F. Douglas, Sci. Adv. **5**, eaav4683 (2019).

[64] G. D'Adamo, R. Menichetti, A. Pelissetto, and C. Pierleoni, Eur. Phys. J. Special Topics **224**, 2239 (2015).

# Chapter 4

# Automated anisotropic coarse-graining of polymers using variational autoencoders

## 4.1 Abstract

We demonstrate the automated coarse-graining of anisotropic molecules and polymers using an autoencoder neural network. The encoder network in an autoencoder is used to automatically generate a latent space that represents the position and orientation of ellipsoidal coarse-grained sites. The decoder network reconstructs an atomistic configuration from the position and orientations encoded in the latent space. This reconstruction from the latent space has a higher fidelity when compared to reconstruction from the center-of-mass alone. This method of automatic anisotropic coarse-graining creates a straightforward strategy to construct an anisotropic coarse-grained representation of semiconducting polymers with anisotropic subunits, and also provides a back-mapping technique that preserves the probability distribution of the conformation space of the original molecule. The automated anisotropic coarse-graining technique is validated through the ability to construct a coarse-grained representation of a solution-phase hexamer of P(NDI2OD-T2), also known as N2200, that can reproduce the physical observable of its atomistic counterpart. The technique is further validated on the comparatively smaller sexithiophene molecule in the liquid phase. We further show that the optimal number of coarse-grained sites can be determined from the loss versus cost for a given number of coarse-grained sites.

## 4.2 Introduction

The recent demand for alternative photovoltaic cells, wearable electronics, and optoelectronic devices have led to intensified research in the area of organic semiconductors.[1–3] This has led to the discovery and utilization of increasingly complex and diverse macromolecules and polymers. It has also become increasingly evident that computational methods, such as molecular dynamics and more recently machine learning, are playing an increasingly large role in material design and discovery.[4–7] However, there are still some limitations on the size and length scale of classical atomistic simulations of materials. Coarse-graining has long been used as a technique to overcome these limitations[8] but to fully utilize a coarse-grained model there needs to be sufficiently accurate, quantifiable, and straightforward back-mapping techniques.

Back-mapping algorithms are important[9] in the field of organic semiconductors because they provide an avenue to study long time-scale properties such as solution-phase aggregation of polymers by running simulations at low resolutions with the possibility of upsampling the system at a later time to study properties such as charge transport or the effects of different functional groups or anisotropy on short-ranged interactions. Many recent breakthroughs in the area of coarse-graining and back mapping came from the integration of machine learning into the field of molecular simulations. The fast-paced growth and development of machine learning tools have increased their popularity in many scientific fields.[10] Autoencoders in particular are popular neural networks developed for data compression problems, in the image-processing sphere,[11] and this machine learning tool has been adapted for uses in the coarse-graining[12] of organic molecules to improve simulation speed and scale. There has been a consistent effort in the attempt to determine the optimal number of coarse-grained sites for generic molecules.[13–16] Unlike traditional methods of coarse-graining, autoencoders do not require a thorough prior understanding of the simulation system, since it is an unsupervised form of machine learning.[17] In general, autoencoders consist of two feedforward neural networks trained together to minimize the data loss between the real data and the data reconstructed from the compressed state. The encoder network is responsible for data compression and in the case of coarse-graining, the encoder network produces the coarse-grained representation of the molecule from the trajectories of the atomistic model obtained from molecular dynamics simulations. On the other hand, the decoder network reconstructs the atomistic trajectory from the coarse-grained representation. This method of coarse-graining attempts to address two major issues in organic semiconductor research. The first is the creation of a coarse-graining methodology that can be compared and optimized without the need for further molecular dynamics simulations. The second problem addressed by the method is its ability to produce a backward map from the coarse-grained representation to the atomistic model.

Even though there have been previous autoencoder models designed to coarse-grain and back-map small molecules to and from isotropic coarse-grain sites,[12] there is still a gap in the knowledge required for coarse-graining macromolecules and polymers into more general ellipsoidal coarse-grain sites accounting for the anisotropy in the mass distribution of different monomers and side-chains.

## 4.3 Theory

For this work, it is assumed that the computational efficiency of a coarse-grain model decreases linearly with the number of sites, and an optimal coarse-grained representation of a molecule is a model which balances computational efficiency with reconstruction fidelity. Since the neural network loss versus the number of coarse-grained sites is defined on the set of integers, it is defined as continuous at point $b$ if $g(b) = f(b)$, where $g(x)$ is a decay curve fit to the data points on the real interval $(a,b)$ and $f(x)$ is a decay curve fit to the data points on the real interval $[b,c)$, the fit is discontinuous otherwise. A given number of coarse-grained sites $b$ is considered optimal on the interval $(a,c)$ if there is a discontinuity at point $b$ as shown in Fig. 4.1. Even though reconstruction fidelity always increases with the number of sites it is expected that for an optimal coarse-grain representation there should be a sharp increase in reconstruction fidelity corresponding to a discontinuity in the loss versus number of site curve.

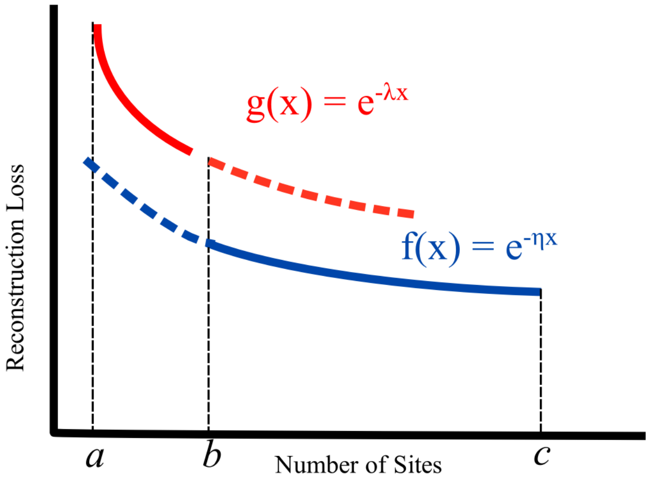


**Figure 4.1:** Diagram showing a typical case where point $b$ is considered an optimal number of coarse-grained sites since $g(b) \neq f(b)$ when $g(x)$ is fitted to the data on the interval $(a,b)$, $f(x)$ is fitted to the data on the interval $[b,c)$, and $g(x)$ and $f(x)$ are both real-valued functions.

A set of mapping functions $\boldsymbol{M}$ are defined such that each fine-grained coordinate $\boldsymbol{r}^n$ is linearly mapped to a unique coarse-grained site $I$ with position $\boldsymbol{R}_I$ and orientation $\boldsymbol{\Omega}_I$ using

$$\boldsymbol{M}_{RI}(\boldsymbol{r}^n) = \boldsymbol{R}_I \tag{4.1}$$

and

$$\boldsymbol{M}_{\Omega I}(\boldsymbol{r}^n) = \boldsymbol{\Omega}_I, \tag{4.2}$$

where $\boldsymbol{M}_{RI}$ maps $\boldsymbol{r}^n$ to the centers-of-mass

$$\boldsymbol{R}_I = \frac{\sum_{i \in \zeta_I} m_i \boldsymbol{r}_i}{\sum_{i \in \zeta_I} m_i}, \tag{4.3}$$

and $\boldsymbol{M}_{\Omega I}$ maps $\boldsymbol{r}^n$ to the principal inertia axes defined by the inertia tensor,

$$\mathbb{I}_{\mathrm{FG},I} = \sum_{i \in \zeta_I} m_i (||\Delta \boldsymbol{r}_i||^2 E - \Delta \boldsymbol{r}_i \Delta \boldsymbol{r}_i^{\mathrm{T}}), \tag{4.4}$$

where $\Delta \boldsymbol{r}_i = \boldsymbol{r}_i - \boldsymbol{R}_I$ is the position of fine-grained particle $i$ relative to the center-of-mass (coarse-grained particle position), $E$ is the $3 \times 3$ identity matrix and the sums are over the set $\zeta_I$ of fine-grained particles that are mapped onto coarse-grained site $I$. For consistency between the coarse-grained and fine-grained models, the configurational distribution of the coarse-grained model must match that of the fine-grained system on which it is based.

### 4.3.1 Data preprocessing

The automatic coarse-graining of polymers can take two possible forms as shown in Fig. 4.2

1. Unconstrained
2. One or more coarse-grained sites per monomer

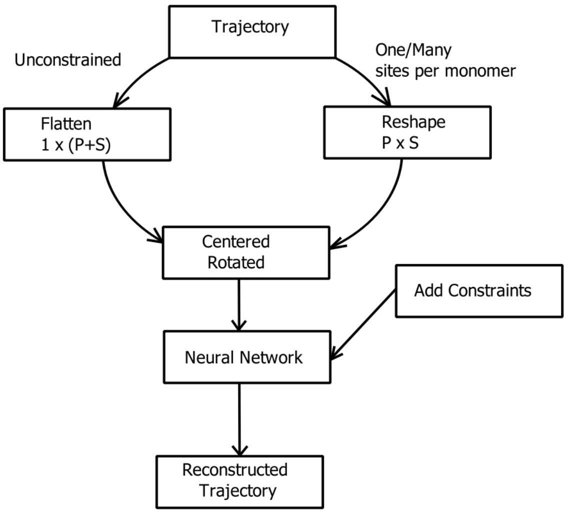


**Figure 4.2:** Schematic of the data processing workflow used to map polymer atomistic configurations to coarse-grained representation.

To determine which method is best suited for a particular polymer, the cost to simulate vs the compression loss must be optimized. In the case of unconstrained coarse-graining, the total number of CG sites is chosen to be less than the number of monomers. The entire polymer is treated as a single macromolecule; that is, for each simulation snapshot, the molecular configuration is flattened into a vector, the center-of-mass is shifted to zero, and the configuration is rotated such that the principal axes of the polymer align with the laboratory frame. The neural network is then unconstrained in allocating atoms to each of the coarse-grained sites. This approach is especially useful for short polymers with simple repeating units. The unconstrained approach can also be used to coarse-grain rigid polymers in which the persistence length is multiple monomers or other cases where it is appropriate to map multiple monomers to a single site. On the other hand, to obtain one or more coarse-grained sites per monomer, the molecular configurations are reshaped to a $P \times S$ matrix, where $P$ is the number of monomer units and $S$ is the number of atoms per monomer, then a similar procedure is followed to center and rotate the polymers in each snapshot with respect to the center-of-mass of each monomer. The neural network is then used to assign a predetermined number of coarse-grained sites to each of the monomer units. For polymers with relatively large repeating units and complex side-chains, it is advantageous to represent the polymer as a $P \times S$ matrix since it increases the number of data points used to train the neural network weights, effectively eliminating the degree of polymerization as a possible source of error.

The neural network method also allows for the integration of prior knowledge into the definition of the coarse-grained sites. A condition can be enforced such that all or some of the coarse-grained sites have the same standard deviation by using the average standard deviation of the specified number of equivalent sites. This condition allows the user to fix the number of CG site types that can be generated independently of the overall number of coarse-grained sites specified. The difference in the reconstruction fidelity as a function of CG site types can also be used to determine the optimal anisotropic coarse-grained representation of any molecule.

### 4.3.2 Encoder algorithm

The encoder network is constructed such that

1. The mass of the coarse-grain site is taken as the sum of the masses of the contributing atoms from the fine-grain model.
2. The inertia tensor of each ellipsoidal site is derived from the average fluctuations of the contributing atoms about the center-of-mass of the coarse-grain site, which will be further explained in the following sections.
3. No atom from the fine-grained model is mapped to more than one coarse-grained site.

The first and second conditions outlined above are satisfied by using Eqns. (4.3) and (4.4) as target values for the construction of a normal distribution with mean $\mu_I$ and standard deviation $\sigma_I$. The mean of the probability distribution of the mass-weighted positions of the atoms corresponds to the mean of the center-of-mass defined in Eqn. (4.3) and the standard deviation of the 3D joint probability distribution of the mass-weighted atom positions generates the principal axes of the coarse-grained site defined by Eqn. (4.4). For the case where more than one center-of-mass is defined corresponding to multiple coarse-grained sites per molecule, the probability distribution becomes a multi-modal distribution. However, straightforward enforcement of the third condition requires no mixing between the modes of the distribution, which would require assigning an atom to the coarse-grained site of the highest probability, according to

$$\boldsymbol{Z_i} = \text{one\_hot}(\underset{I}{\text{argmax}}\{\log \pi_{iI}\}) \tag{4.5}$$

where $\boldsymbol{Z_i}$ is a categorical variable and $\pi_{iI}$ is the probability that atom $i$ is assigned to coarse-grain site $I$. However, this argmax function would make the neural network nondifferentiable and prevent learning through backward propagation.[18] To enforce the first condition without trying to backpropagate through a non-differentiable layer, the Gumbel-softmax reparametrization[18] trick is used to approximate an argmax function. Gumbel-softmax reparameterization allows a variational autoencoder to approximate sampling from a discrete latent space through the introduction of a neural network temperature variable giving the $I$th element of $\boldsymbol{Z}_i$ as

$$Z_{iI} = \frac{\exp((G_{iI} + \log \pi_{iI})/\tau)}{\sum_j^n \exp((G_{jI} + \log \pi_{jI})/\tau)} \tag{4.6}$$

Here, $G_{iI}$ is a sampled from the standard Gumbel distribution and $\tau$ is the temperature variable, such that as $\tau \to 0$ the softmax calculations smoothly approach argmax and $\boldsymbol{Z}_i$ approximates a one-hot vector. By initializing the neural network with a sufficiently large temperature variable, each atom in a molecule can transition across all available coarse-grained sites.[12] The subsequent annealing process lowers the temperature gradually ensuring that each atom is mapped to the optimal coarse-grained site in such a way that the overall coarse-grained model reproduces the mass distribution of the all-atom model. The encoder network performs linear transformations assigning the atomistic configurations to the centers-of-mass and the inertia tensor of the coarse-grained ellipsoid. By retaining the mass distribution along each of the principal axes, the model generalizes spherically symmetric coarse–grain sites to anisotropic ellipsoidal sites. As the neural network temperature variable decreases, each atom only contributes to the calculation of the mean of a single coarse-grained site and the fluctuation of the atom about the mean position defines the standard deviation and by extension the principal axes of the coarse-grained site. Since each molecular trajectory is fixed to the molecular center-of-mass, atoms close to the molecular center-of-mass will have smaller fluctuations and will be the first to anneal into their final position. Atoms at the far ends of a polymer or side-chains will fluctuate more widely and will require more data to produce consistent results for their coarse-grained representation. The latent space of the encoder network provides a set of positions $\boldsymbol{R}_I$ and orientations $\boldsymbol{\Omega}_I$ for each coarse-grained site.

### 4.3.3 Decoder algorithm

The decoder is responsible for the reconstruction of the atomistic trajectories from the coarse-grained latent space representation.[19] In the automatic anisotropic coarse-graining method, the reconstruction is done using two pieces of information, the center-of-mass of each coarse-grained site as well as the inertia tensor which describes an ellipsoidal mass distribution about each of the coarse-grained center-of-mass. Compared to a spherical coarse-grained model, reconstruction fidelity is improved for the anisotropic model since it uses information about the inertia tensor in the decoding process. This additional reconstruction fidelity is important for organic semiconductors since back mapping is an important tool to understand charge transfer in polymer aggregates.[20]

The loss function of the autoencoder has two sources contributing to the total loss, The first being the reconstruction loss and the second being the reparameterization loss. The reconstruction loss can be further broken down into the reconstruction of one–, two– and three–body contributions, that is, the reconstruction of the atom positions, bonds, and angles respectively. This is achieved through the use of a regression loss function namely, the mean squared error,

$$L_{\text{recon}} = \|\Gamma_{\text{D}}(\Gamma_{\text{E}}(\boldsymbol{X}, \tau, \boldsymbol{G})) - \boldsymbol{X}\|^2 \tag{4.7}$$

where $\Gamma_{\text{D}}$ and $\Gamma_{\text{E}}$ are the decoder and encoder network function, and $\tau$ and $\boldsymbol{G}$ are the neural network temperature variable and the sampled Gumbel distribution, respectively. On the other hand, since the reconstruction of the trajectories is probabilistic, the reparametrization error minimizes the distance between the true distribution of the atomic positions and the sampled distribution used for the reconstruction. This reparameterization error is constructed as the evidence lower bound.[21] The variational autoencoder aims to maximize the likelihood of recovering the data from the latent representation, $p(\boldsymbol{Z}|\boldsymbol{X})$, where $\boldsymbol{Z}$ is the latent representation and $\boldsymbol{X}$ is the data. given the input data has true distribution $p(\boldsymbol{X})$ and the latent representation has distribution $q(\boldsymbol{Z})$, the evidence lower bound is defined as

$$\text{ELBO} = \mathbb{E}_q \left[ \log \frac{p(\boldsymbol{X}|\boldsymbol{Z})}{q(\boldsymbol{Z})} \right], \tag{4.8}$$

where $\mathbb{E}_q$ is the expectation. The total loss is calculated using

$$L_{\text{total}} = L_{\text{recon}} - 0.5 \times \text{ELBO} \tag{4.9}$$

The gradient descent algorithm was implemented using the Adam optimizer.[22] A schematic of the autoencoder architecture is shown in Fif. 4.3.

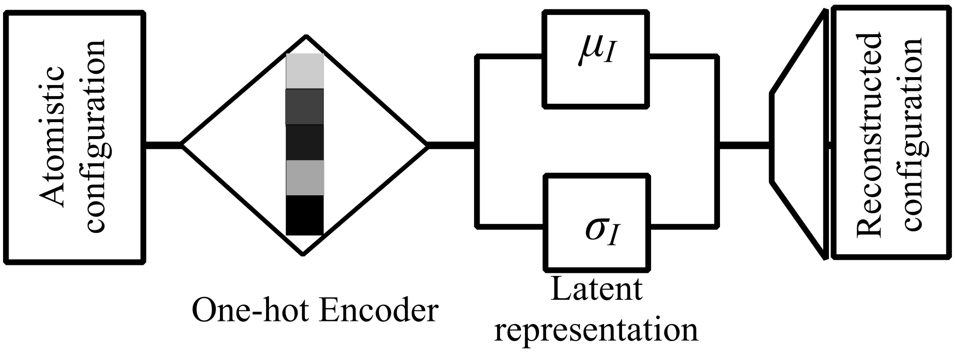


**Figure 4.3:** Schematic of the neural network architecture used to map polymer atomistic configurations to a discrete latent space parameterized by the mean and standard deviation of a multimodal joint ellipsoidal distribution.

### 4.3.4 Atomistic simulation and coarse-grained potential

$\alpha$-Sexithiophene and a hexamer of the polymer P(NDI2OD-T2), also known as N2200, were chosen as test molecules to demonstrate the different capabilities of the anisotropic autoencoder. Atomistic MD simulations were done using the molecular dynamics software package LAMMPS (version 20NOV19).[23–25] The OPLS-AA force field[26–29] and a cut-off of 10 Å were used for the simulation of 250 sexithiophene molecules in the isothermal-isobaric (NPT) ensemble with the pressure set at 1 atm and temperature of 680 K.[30] A molecular dynamics simulation of a single N2200 hexamer in a solution of 14680 chloroform molecules was carried out at 300 K and 1 atm in the NPT ensemble with OPLS-AA force field and a cutoff of 11 Å. For all atomistic simulations hydrogen bonds were constrained with the SHAKE algorithm,[31] long-ranged electrostatic interactions were calculated with the particle–particle particle–mesh (PPPM) method,[32,33] and the temperature and pressure controlled by a Nosé–Hover thermostat and barostat.[34,35] The N2200 hexamer in chloroform solution was equilibrated for 1 ns and then simulations were carried out with the time step set to 2 fs and simulations ran for 1 ns. The sexithiophene simulations were 25 ns long with a timestep of 1 fs. The last 20 ns of the simulation data was used for parameterization of the coarse-grained potential and calculation of structural distributions.

To use the coarse-grained models for molecular dynamics simulations, the coarse-grained potential was fitted by using the instantaneous forces and torques to train a neural network potentntial with explicit inclusion of dihedral angles between nearest-neighbor anisotropic monomers. This corresponds to the force-matching condition in the AFM-CG method, which is required for thermodynamic consistency. A schematic of the neural network used in the force matching procedure is shown in Fig. 4.4.

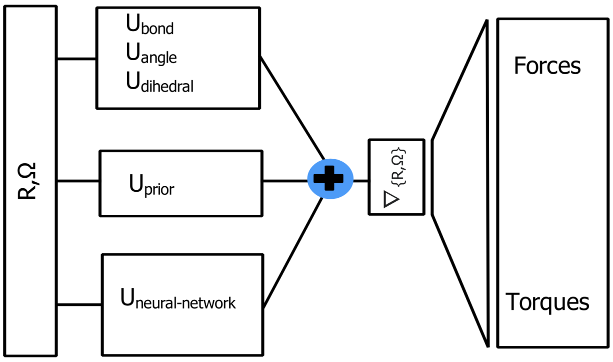


**Figure 4.4:** Schematic of the neural network used to fit the coarse-grain potential.

Each monomer had a ghost atom attached at off-center positions for the definition of bonds between ellipsoids Fig. 4.5. This ensures that forces and torques are correctly applied to the anisotropic particle and not just the center-of-mass of the monomer. The bond length, bond angle, and dihedral potentials are given by

$$U_{\mathrm{bond}} = K_{\mathrm{B}}(b - b_0)^2 \tag{4.10}$$

$$U_{\mathrm{angle}} = K_{\mathrm{A}}(\theta - \theta_0)^2 \tag{4.11}$$

$$\begin{aligned} U_{\mathrm{dihedral}} &= \frac{1}{2}K_1[1+cos(\phi)] + \frac{1}{2}K_{\mathrm{D}}[1-cos(2\phi)] \\ &\quad + \frac{1}{2}K_3[1+cos(3\phi)] + \frac{1}{2}K_4[1-cos(4\phi)] \end{aligned} \tag{4.12}$$

where $b$ and $b_0$ are the instantantous and equilibrium bond lengths, respectively, $\theta$ and $\theta_0$ are the instantaneous and equilibrium bond angles, respectively, $\phi$ is the dihedral angle, $K_{\mathrm{B}}$ and $K_{\mathrm{A}}$ are the bond and three-body angle potential parameter, respectively, and $K_1$, $K_{\mathrm{D}}$, $K_3$, and $K_4$ are the coefficient of the OPLS cosine expansion of the dihedral potential. Non-bonded interactions were defined between particles separated by one bond (1–2 interactions).

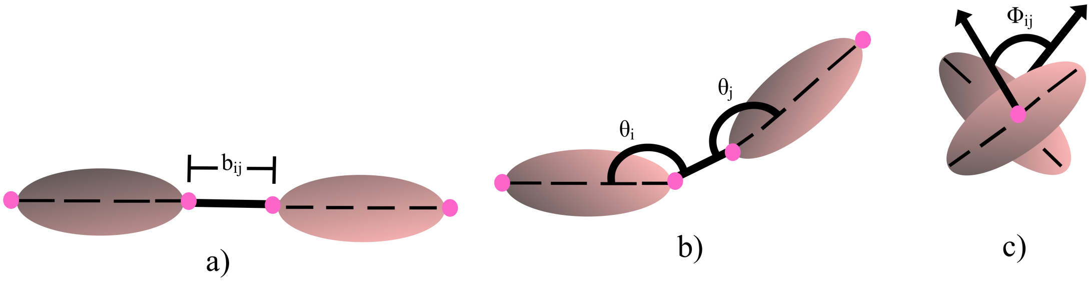


**Figure 4.5:** Schematic of the (a) bonds, (b) angles, and (c) dihedrals as defined for the anisotropic polymer models.

Fitting the forces to the derivative of the potential was done using TensorFlow's gradient descent algorithm and the derivative of the potential was implemented using TensorFlow's GradientTape function to evaluate the computational derivative.[36,37] The hyperparameters for the neural network was fitted using modified a modified version of the Behler symmetry functions.[38] When fitting the coarse-grained potential using the neural network, each coarse-grain site is mapped to an invariant vector representation $\boldsymbol{D}_{IJ}$ which is defined in terms of the position and orientations of particles $I$ and $J$ and is given by

$$\begin{aligned}\boldsymbol{D}_{IJ} = \{ & R_{IJ}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,1}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,2}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,3},\\ & \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,3},\\ & \boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,3},\\ & \boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,3},\\ & \boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,3}\},\end{aligned} \tag{4.13}$$

where $\boldsymbol{R}_I$, $\boldsymbol{R}_J$, $\boldsymbol{\Omega}_I$, and $\boldsymbol{\Omega}_J$ are obtained from the encoder latent space and $\boldsymbol{R}_{IJ} \equiv \boldsymbol{R}_I - \boldsymbol{R}_J$. The neighbourhood of particle $I$ can then be represented by a unique fingerprint $\mathbb{D}_I$ which is obtained from the concatenation of all the $\boldsymbol{D}_{IJ}$ vectors in the neighbourhood of particle $I$. The prior repulsive potential can then be represented by the simply as

$$U_{\text{prior},I} = \sum_{J\neq I} B_1 \sigma_{\text{c}} \left(\mathbb{D}_I\right)^{-B_2}, \tag{4.14}$$

where $\sigma_{\text{c}}$ is a neural-network function and $B_1$ and $B_2$ are trainable parameters. The total potential $U$ can then be written as a sum over all $U_I$ contribution given as

$$U_I = U_{\text{NN},I} + U_{\text{prior},I} + U_{\text{bond},I} + U_{\text{angle},I} + U_{\text{dihedral},I}. \tag{4.15}$$

and

$$U = \sum_{I=1}^{N} U_I \tag{4.16}$$

A more indepth discussion of the force matching neural network architecture can be found in the supporting information. The interaction between the N2200 ellipsoids and the spherical solvent particles as well as the solvent–solvent interactions were derived from the same procedure above.

A six-site coarse-grained representation was used for the coarse-grained simulation of both N2200 and sexithiophene. The simulations were done in the canonical ensemble (NVT) to match the density of the atomistic simulations. The sexithiophene coarse-grained simulations were performed at 590 and 680 K with 250 molecules. A single-site model of sexithiophene was also parameterized under the same conditions. The CG simulations with the N2200 hexamer and 14680 isotropic chloroform solvents were done at 300 K.[39] The N2200 hexamer in chloroform solution as well as the sexitiophene coarse-grained simulations were 25 ns long with the last 20 ns used for the calculation of structural distributions.

## 4.4 Results and Discussion

### 4.4.1 $\alpha$-Sexithiophene

Sexithiophene shown in Fig. 4.6 has been researched as a promising material for organic photovoltaics[40] and organic light-emitting diodes.[41,42] There has been significant research into controlling the orientation of sexithiophene deposited on substrates.[43,44] Sexithiophene was used to demonstrate the unconstrained coarse-graining ability of the anisotropic autoencoder. Sexithiophene coarse-grained to a single ellipsoid does not capture the backbone flexibility or any of the thiophene-thiophene torsional configurations. The neural network loss was calculated for different numbers of coarse-grained sites ranging from one to six. The plot of loss versus the number of sites in Fig. 4.7 shows a notable decrease in reconstruction loss when more than one coarse-grained site is used to model sexithiophene, whereas there is a smaller decrease in reconstruction loss when the number of sites increases from two to six. Since sexithiophene consists of six monomers, a steep decrease in the neural network reconstruction loss between five and six coarse-grain sites is expected. With six available coarse-grain sites the neural network can more accurately reconstruct the mass distribution changes due to the rotation of the monomers about the thiophene–thiophene bonds as shown in Fig. 4.8.

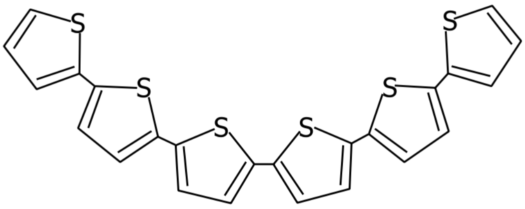


**Figure 4.6:** Chemical structure of sexithiophene

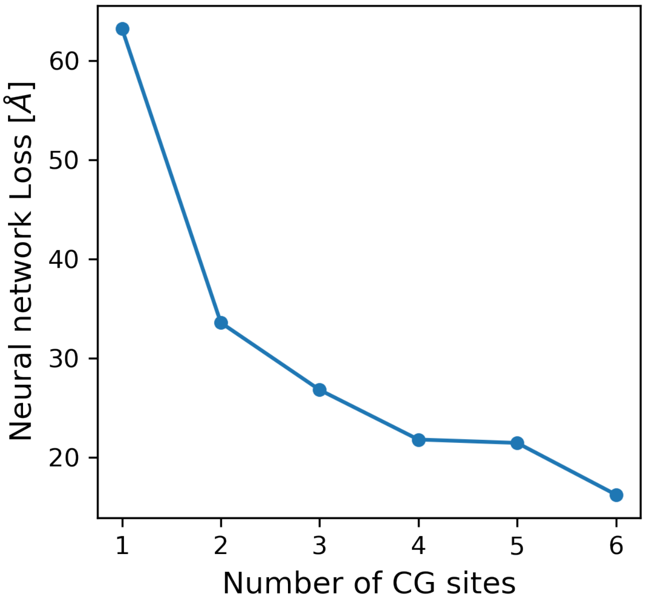


**Figure 4.7:** Neural network reconstruction loss versus the number of coarse-grained sites when sexithiophene is mapped to between one and six coarse-grain sites.

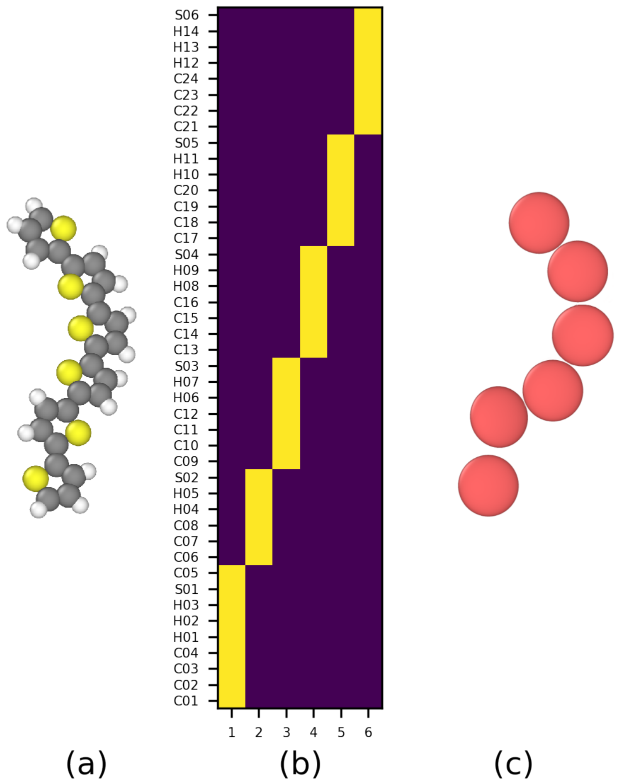


**Figure 4.8:** Six-site coarse-grained model of sexithiophene. The (a) atomistic configuration was mapped to the latent space using the (b) learned encoding, producing a mapping to the (c) position and orientation of the coarse-grained sites. The rows of the encoding matrix in (b) represent each atom and the columns are the available coarse-grained sites.

The six-site neural network model of sexithiophene captures the structural variations in the liquid and liquid crystal phases as shown in Fig. 4.9. The six-site coarse-grain model of sexithiophene outperforms the single-site model when comparing the orientational order parameter in the liquid crystal phase (Fig. 4.10). However, there are only small differences between the six-site and the single-site model when comparing the center of mass radial distribution function (Fig. 4.11).

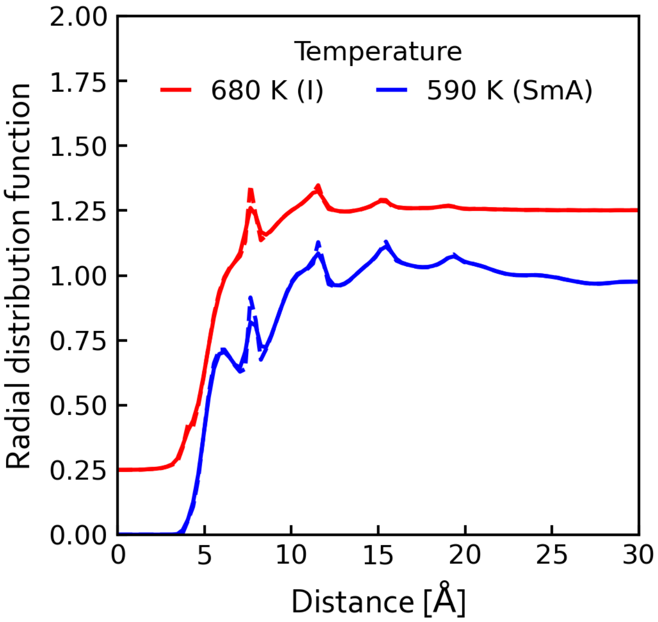


**Figure 4.9:** All-atom (solid lines) and six-site (dashed lines) coarse-grained monomer-monomer radial distribution function for sexithiophene in the isotropic phase (680 K) and the Smectic-A phase (590 K).

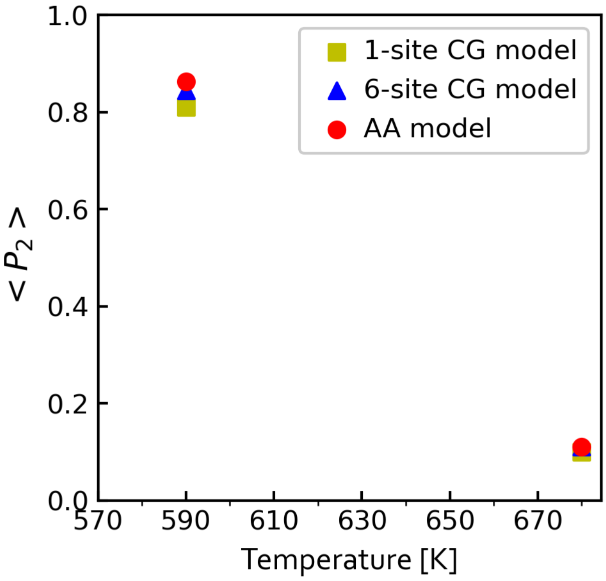


**Figure 4.10:** Orientational order parameter for all-atom, six-site coarse-grain and one-site coarse-grain models of sexithiophene at 590 and 680 K

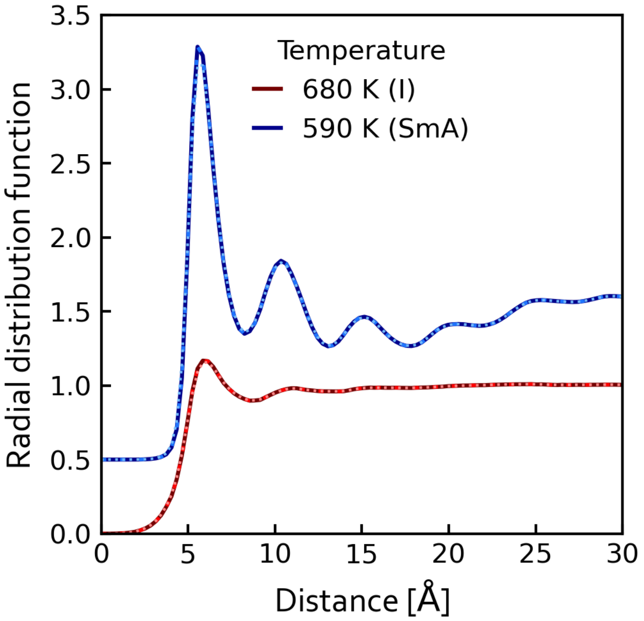


**Figure 4.11:** All-atom (solid lines), six-site coarse-grained (dashed lines), and single-site coarse-grained (dotted lines) center-of-mass radial distribution function for sexithiophene in the isotropic phase (680 K) and the smectic-A phase (590 K).

The one-site sexithiophene model had a 132 × speed-up compared to the all-atom model while the six-site sexithiophene model had a 17 × speed-up compared to the all-atom model.

### 4.4.2 P(NDI2OD-T2)

Poly[N,N′-bis(2-octyldodecyl)naphthalene-1,4,5,8-bis(dicarboximide)-2,6-diyl]-alt-5,5′-(2,2′-bithiophene) (P(NDI2OD-T2)), also known as N2200, is a copolymer of naphthalene diimide (NDI) and bithiophene units with alkyl side chains. There have been significant interest in N2200 as an organic semiconductor.[45–49] It is considered one of the best organic polymer acceptors due to its high electron mobility[50] and narrow band gap.[51] N2200 has had recent success in organic solar cell applications[52] and energy storage in the form of capacitors.[46] N2200 was chosen to demonstrate how well the anisotropic autoencoder handles one or more coarse-grained sites per monomer. This also provides an opportunity to see how well the neural network method handles flexible alkyl side chains and an aromatic backbone. Plots of the neural network loss versus the number of coarse-grained sites are shown in Fig. 4.12. These plots showed several discontinuities where the loss between consecutive numbers of coarse-grained sites showed a larger decrease than for the pair before or the pair after.

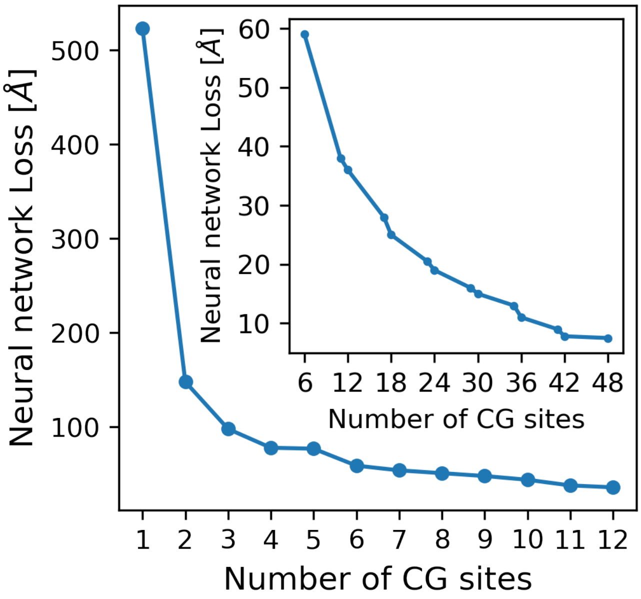


**Figure 4.12:** Neural network reconstruction loss versus number of coarse-grained sites for the case where the number of sites is less than twice the number of monomers for the N2200 hexamer (main plot) and for one or more sites per monomer for the N2200 hexamer. Lines connecting the data points are solely for visualizing the trend between adjacent data points.

The unconstrained allocation of coarse-grained sites for the N2200 hexamer starts with a relatively high error which can be attributed to the attempt to represent a flexible molecule as a rigid ellipsoid. Even though there is a significant drop in the reconstruction error between one and five sites, the trend still follows the expected exponential decay that would be expected just from adding more complexity to the model. The only significant feature that is observed on the interval $[1, 12]$ is a discontinuity in the decay trend between five and six coarse-grained sites. As expected there is a significant drop in the reconstruction error when each of the monomers in the polymer is assigned to individual sites. There is a discontinuity in the plot of reconstruction loss versus the number of sites when eighteen and forty-two coarse-grained ellipsoids are allocated. The coarse-grained model with three sites per monomer separated the backbone of the polymer from the sidechains. The allocation of the atoms associated with forty-two sites or seven sites per monomer is shown in Fig. 4.13. The anisotropic autoencoder was able to group each branch of the alkyl side chains into an ellipsoid while also grouping the naphthalene diimide (NDI) and bithiophene units into individual ellipsoids. Fig. 4.12(inset) shows a trend of increased reduction in the loss for every six additional sites added to the polymer. By observing how the neural network allocates the atoms for the seven-site model, a priori information can be added to the neural network by enforcing a set of only three unique ellipsoid types for the seven available sites: that is, four ellipsoids assigned to the first type, two to the second and one to the third. The results are shown in the color-coding of Fig. 4.13. This added flexibility can significantly simplify the output of the neural network latent space with less than 2 % increase in the reconstruction error.

The comparison of the center-of-mass radius of gyration for the all-atom, six-site coarse-grained, and back-mapped models is shown in Fig. 4.14. There is a close match between the all-atom and the back-mapped models, the discrepancy between the coarse-grained and all-atom models can be attributed to the standard way of calculating the radius of gyration of a polymer, which has not been modified to account for anisotropic particles.

The method of calculation, assumed that the entire mass of each monomer acts at the center-of-mass of the coarse-grain ellipsoid instead of distributed over the entire volume. However, the back-mapped model gives a better representation of the radius of gyration of the coarse-grained model.

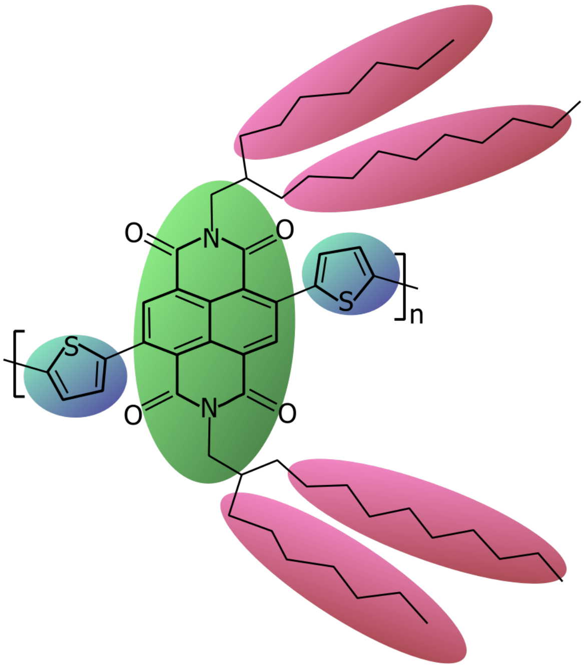


**Figure 4.13:** Neural network representation of coarse-grained N2200 where the number of coarse-grained sites is set to seven disjoint sets and the color- coding represents sites with the same inertia tensor.

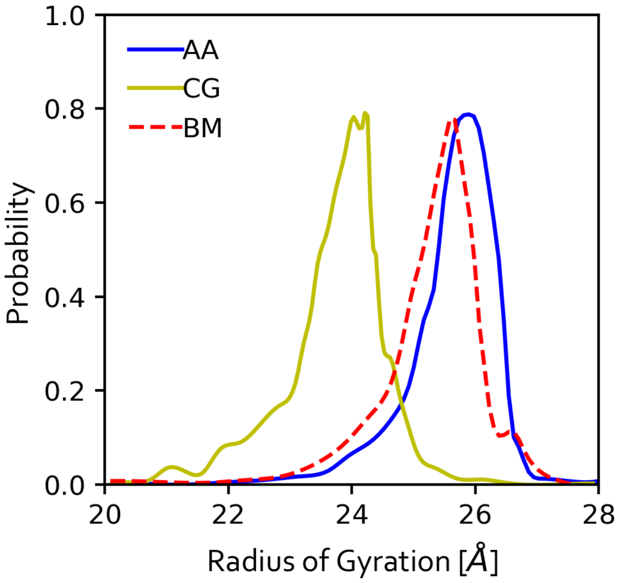


**Figure 4.14:** A comparison of (a) the distribution of the center-of-mass radius of gyration of the all-atom (AA), six-site coarse-grained (CG), and the back mapped (BM) model of the N2200 hexamer in chloroform solution at 300 K.

The six-site representation of N2200 hexamer provides an opportunity for high fidelity backmapping. The six-site CG model in which each monomer is coarse-grained into a single site does not capture the structural correlations very well, and has a stronger effective monomer-solvent repulsion than the all-atom model as shown in the monomer-solvent center-of-mass radial distribution function in Fig. 4.15. The six-site model A higher resolution model with separate sites for backbone and side-chains is likely needed to accurately capture the radial distribution function. When the flexible side-chains are coarse-grained together with the backbone, the CG site dimensions become more isotropic due to an increase in size along the $\pi$-stacking direction. This limitation of the six-site coarse-grained model can explain the discrepancy between the all-atom and coarse-grained radial distribution function.

The six-site N2200 model had a 161 × speed-up compared to the all-atom model

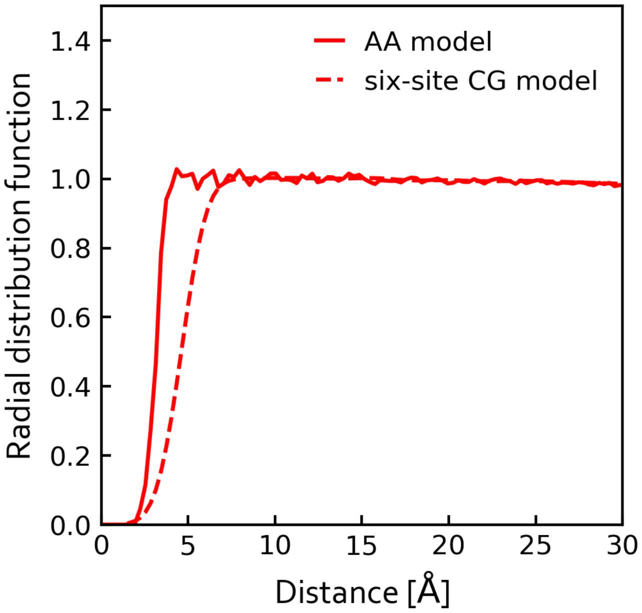


**Figure 4.15:** The monomer-solvent center-of-mass radial distribution function for the all-atom and six-site coarse models of the N2200 hexamer in chloroform solution at 300 K.

# CONCLUSIONS

We have shown that an unsupervised machine-learning approach can be used to coarse-grain large molecules and polymers using either an unconstrained approach or by prescribing one or more sites per monomer. With the inclusion of anisotropic mass distribution data for the coarse-grained sites, the autoencoder was able to increase the reconstruction fidelity of large molecules with anisotropic mass distribution. The anisotropic feature is especially highlighted with the organic semiconducting polymer sexithiophene and N2200 since they both contain anisotropic monomer units. Additionally, the automatic anisotropic coarse-graining method provides the ability to specify the number of unique types of ellipsoids independently of the specified number of coarse-grained sites. This feature simplifies the coarse-grained representation of polymers with complex monomers such as N2200.

# References

[1] P. Meredith, W. Li, and A. Armin, Adv. Energy Mater. **10**, 2001788 (2020).

[2] X. Yang, L. Ding, *et al.*, J. Semicond **42**, 090201 (2021).

[3] K. Yu, S. Rich, S. Lee, K. Fukuda, T. Yokota, and T. Someya, Proc. IEEE **107**, 2137 (2019).

[4] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, and M. Lei, InfoMat **1**, 338 (2019).

[5] C. H. Chan, M. Sun, and B. Huang, EcoMat , e12194 (2022).

[6] S. Xiao, R. Hu, Z. Li, S. Attarian, K.-M. Björk, and A. Lendasse, Neural Comput. App. **32**, 14359 (2020).

[7] T. Okamoto, S. Kumagai, E. Fukuzaki, H. Ishii, G. Watanabe, N. Niitsu, T. Annaka, M. Yamagishi, Y. Tani, H. Sugiura, *et al.*, Sci. Adv. **6**, eaaz0632 (2020).

[8] J. Jin, A. J. Pak, A. E. Durumeric, T. D. Loose, and G. A. Voth, J. Chem. Theory Comput. **18**, 5759 (2022).

[9] W. Li, C. Burkhart, P. Polińska, V. Harmandaris, and M. Doxastakis, J. Chem. Phys. **153**, 041101 (2020).

[10] K. K. Bejagam, S. Singh, Y. An, and S. A. Deshmukh, J. Phys. Chem. Lett. **9**, 4667 (2018).

[11] Y. Zhang, in *ICONIP17-DCEC. Available online: http://users. cecs. anu. edu. au/Tom. Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58. pdf (accessed on 23 March 2017)* (2018).

[12] W. Wang and R. Gómez-Bombarelli, npj Comput. Mater. **5**, 1 (2019).

[13] A. V. Sinitskiy, M. G. Saunders, and G. A. Voth, J. Phys. Chem. B **116**, 8363 (2012).

[14] M. Li, J. Z. Zhang, and F. Xia, J. Chem. Theory Comput. **12**, 2091 (2016).

[15] Z. Wu, Y. Zhang, J. Z. Zhang, K. Xia, and F. Xia, J. Comput. Chem. **41**, 14 (2020).

[16] A. E. P. Durumeric and G. A. Voth, J. Chem. Phys. **151**, 124110 (2019).

[17] C. Doersch, arXiv preprint arXiv:1606.05908 (2016).

[18] E. Jang, S. Gu, and B. Poole, arXiv (2016).

[19] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, arXiv (2017).

[20] N. Rolland, M. Modarresi, J. F. Franco-Gonzalez, and I. Zozoulenko, Comput. Materi. Sci. **179**, 109678 (2020).

[21] A. Asperti and M. Trentin, IEEE Acc. **8**, 199440 (2020).

[22] D. P. Kingma and J. Ba, arXiv , 1412.6980 (2014).

[23] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[24] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington, Comput. Phys. Commun. **183**, 449 (2012).

[25] W. M. Brown, P. Wang, S. J. Plimpton, and A. N. Tharrington, Comput. Phys. Commun. **182**, 898 (2011).

[26] W. L. Jorgensen and J. Tirado-Rives, J. Am. Chem. Soc. **110**, 1657 (1988).

[27] R. C. Rizzo and W. L. Jorgensen, J. Am. Chem. Soc. **121**, 4827 (1999).

[28] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

[29] W. L. Jorgensen and N. A. McDonald, J. Mol. Struct. THEOCHEM **424**, 145 (1998).

[30] A. Pizzirusso, M. Savini, L. Muccioli, and C. Zannoni, J. Mater. Chem. **21**, 125 (2011).

[31] J.-P. Ryckaert, G. Ciccotti, and H. J. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[32] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (CRC Press, 1998).

[33] M. L. Price, D. Ostrovsky, and W. L. Jorgensen, J. Comput. Chem. **22**, 1340 (2001).

[34] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[35] S. Nosé, Mol. Phys. **52**, 255 (1984).

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, in *12th USENIX Symp. Oper. Syst. Des. Implement. OSDI 16* (2016) pp. 265–283.

[37] F. Chollet *et al.*, “Keras,” https://github.com/fchollet/keras (2015).

[38] J. Behler, J. Chem. Phys. **134**, 074106 (2011).

[39] B. J. Boehm, C. R. McNeill, and D. M. Huang, Nanoscale **14**, 18070 (2022).

[40] J. Sakai, T. Taima, and K. Saito, Org. Elect. **9**, 582 (2008).

[41] C. Heck, T. Mizokuro, M. Misaki, R. Azumi, and N. Tanigaki, Jap. J. App. Phys. **50**, 04DK20 (2011).

[42] C. Heck, T. Mizokuro, and N. Tanigaki, App. Phys. Exp. **5**, 022103 (2012).

[43] T. Matsushima and H. Murata, App. Phys. Let. **98**, 121 (2011).

[44] T. Mizokuro, C. Heck, and N. Tanigaki, J. Phys. Chem.B **116**, 189 (2012).

[45] J. Yuan, W. Guo, Y. Xia, M. J. Ford, F. Jin, D. Liu, H. Zhao, O. Inganäs, G. C. Bazan, and W. Ma, Nano Energy **35**, 251 (2017).

[46] B. A. Wavhal, M. Ghosh, S. Sharma, S. Kurungot, and S. Asha, Nano. **13**, 12314 (2021).

[47] G. Wen, X. Zou, R. Hu, J. Peng, Z. Chen, X. He, G. Dong, and W. Zhang, RSC Adv. **11**, 20191 (2021).

[48] Y. Yan, Y. Liu, J. Zhang, Q. Zhang, and Y. Han, J. Materi. Chem. C **9**, 3835 (2021).

[49] C. Ren, Y. He, S. Li, Q. Sun, Y. Liu, Y. Wu, Y. Cui, Z. Li, H. Wang, Y. Hao, *et al.*, Org. Elect. **70**, 292 (2019).

[50] H. Yan, Z. Chen, Y. Zheng, C. Newman, J. R. Quinn, F. Dötz, M. Kastler, and A. Facchetti, Nat. **457**, 679 (2009).

[51] C. Mu, P. Liu, W. Ma, K. Jiang, J. Zhao, K. Zhang, Z. Chen, Z. Wei, Y. Yi, J. Wang, *et al.*, Adv. Mater. **26**, 7224 (2014).

[52] K. Wang, S. Dong, K. Zhang, Z. Li, J. Huang, and M. Wang, Org. Elect. **99**, 106319 (2021).

# Chapter 5

# Automatic labeling and prediction of anisotropic semiflexible polymers aggregate phase diagrams using neural networks

## 5.1 Abstract

A machine learning pipeline has been developed to understand the role of polymer backbone flexibility in the temperature-dependent aggregation behavior of anisotropic polymers. A toy polymer model is used to conduct simulations with variations in a predefined set of polymer properties. The set of variable properties used to model polymer backbone flexibility includes the coefficient of the angle potential and the coefficient of the dihedral potential. The temperature of the simulation is also used as a variable to determine the effect of temperature on the polymer conformations observed. The machine-learning pipeline developed was able to assign an aggregate type to unlabelled polymer trajectories as well as predict the type of aggregate based on the predefined properties of the polymer interaction potential.

## 5.2 Introduction

Organic semiconducting polymers, which typically consist of highly anisotropic monomers, are a major area of focus in the search for cheap, flexible, and printable optoelectronic devices such as light-emitting diodes and photovoltaic cells.[1–3] The ability to tune the polymer's flexibility and solubility makes them ideal for solution processing.[4–6] However, to maximize charge transport and overall device efficiency, a deeper understanding of the polymer aggregation process and the drivers of this process is needed.[7] The charge transport capabilities of an organic semiconductor are affected by chain size, persistent length, and overall crystallinity of the polymer.[8] mesoscopic features such as crystallinity and grain sizes are further driven by molecular properties such as the dihedral angle between monomers and processing conditions such as temperature and annealing rates[9]. To fully conceptualize the design space of organic semiconducting devices, mesoscopic polymer aggregation predictions must be able to consider both molecular properties and processing conditions.

Computational approaches such as molecular dynamics simulations play an important role in bridging the atomistic and mesoscopic length scales.[10] However, atomistic simulations of bulk polymer aggregates on equilibrium time scales are not feasible. To bridge the gap between atomistic and mesoscopic time scales, coarse-grained (CG) simulations are often used.[11] It is however important to note that, anisotropic polymers are best represented by anisotropic subunits capable of capturing the $\pi$–$\pi$ stacking configuration between polymers using a single CG site.[12] To this end, a significant amount of research has gone into the development of anisotropic potentials and coarse-grained models[13–16] capable of reproducing the bonds, angles, and dihedral distributions of the polymer backbone and side chains. These CG models allow for the efficient sampling of the conformational space of anisotropic polymers by tuning the backbone flexibility.

The conformational space of polymer organic semiconductors is a high-dimensional space with highly complex relationships between parameters. Machine learning has been effective in processing data from high-dimensional data sets while providing useful insight into the complex relationship between input and target variables.[17] There have been significant advances in the accessibility of machine learning to design powerful architectures with off-the-shelf layers and functions.[18] It is especially easy to design variational autoencoders for dimensionality reduction problems and feedforward classification networks which are useful in grouping large amounts of data into predefined disjoint sets.[19] There have been previous attempts at using non-machine learning approaches to predict the aggregation behavior of semiflexible polymers with strictly isotropic monomers.[20] Previous works, also explored the aggregation phase diagram of semiflexible polymers using molecular dynamics simulations without predictive capabilities.[21] Machine-learning approaches have been explored with great success, especially in the field of computational biology.[22]

In this work, we develop two data-driven workflows assisted by machine learning to identify, classify and predict the types of polymer aggregates obtained from simulating anisotropic polymers with varying properties under different simulation conditions. The first algorithm uses an autoencoder to subdivide the entire conformational space of the simulated polymer into a predefined number of disjoint sets that can be easily labeled manually. The second algorithm attempts to predict the most probable polymer aggregate to form under specific simulation conditions for a given set of molecular scale polymer properties. Together, these algorithms are capable of combining molecular scale properties and processing conditions to predict the mesoscopic bulk behavior of polymer aggregates and potentially inform design choices for organic optoelectronic devices.

## 5.3 Theory

### 5.3.1 Anisotropic polymer model

The generalized coarse-grained polymer model and procedure used for the simulations have been fully described in previous works.[23] These coarse-grained polymers have been designed with the Gay-Berne biaxial potential for dissimilar particles[24,25] and explicit inclusion of dihedral angles between nearest-neighbor anisotropic monomers. The anisotropic Gay-Berne potential is implemented in the LAMMPS package[26] and is given by the expression[12]

$$\begin{aligned} U_{\mathrm{GB}}(\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{r}_{12}) &= U_r(\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{r}_{12},\gamma)\cdot\eta_{12}(\boldsymbol{A}_1,\boldsymbol{A}_2,\nu)\cdot \\ &\quad \chi_{12}(\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{r}_{12},\mu) \end{aligned} \tag{5.1}$$

where

$$U_r = 4\varepsilon(\rho^{12} - \rho^6) \tag{5.2}$$

$$\rho = \frac{\sigma}{h_{12} + \gamma\sigma} \tag{5.3}$$

where $\boldsymbol{r}_{ij}$ is the distance between the centers-of-mass of the two ellipsoids, $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$ are the rotation matrices transforming the orientation of the ellipsoids from lab frame to body frame. $h_{12}$ is the approximation to the distance of closest approach and $\gamma$ and $\mu$ are both set to 1.0. Reduced LJ units are used, so lengths are in units of $\sigma$, energy in units of $\varepsilon$ and temperature in units of $\varepsilon/k_{\mathrm{B}}$ . The mass is in units of the monomer mass $m$ and time is in units of $\sqrt{m\sigma^2/\varepsilon}$.[12] Each monomer has noninteracting "ghost" atoms attached at off-center positions for the definition of bonds between ellipsoids. This ensures that forces and torques are correctly applied to the anisotropic particle and not just the center of mass of the monomer. The polymer semiflexibility and dihedral barrier height are determined by the following equations for the bond length, bond angle, and dihedral angle potentials,[12]

$$E_{\mathrm{bond}} = K_{\mathrm{B}}(b - b_0)^2, \tag{5.4}$$

$$E_{\mathrm{angle}} = K_{\mathrm{A}}(\theta - \theta_0)^2, \tag{5.5}$$

$$\begin{aligned} E_{\mathrm{dihedral}} &= \frac{1}{2}K_1[1 + cos(\phi)] + \frac{1}{2}K_{\mathrm{D}}[1 - cos(2\phi)] \\ &\quad + \frac{1}{2}K_3[1 + cos(3\phi)] + \frac{1}{2}K_4[1 - cos(4\phi)], \end{aligned} \tag{5.6}$$

where $b$ and $b_0$ are the instantantous and equilibrium bond lengths, respectively, $\theta$ and $\theta_0$ are the instantaneous and equilibrium bond angles, respectively, $\phi$ is the dihedral angle, $K_{\mathrm{B}}$ and $K_{\mathrm{A}}$ are the bond and three-body angle potential parameter, respectively, and $K_1$, $K_{\mathrm{D}}$, $K_3$, and $K_4$ are the coefficient of the OPLS cosine expansion of the dihedral potential.

The angle coefficient $K_{\mathrm{A}}$ and the second coefficient of the OPLS cosine expansion $K_{\mathrm{D}}$ are manipulated to represent various backbone flexibility of typical organic semiconductors. For this work the length of the polymer chain was varied between 22 and 64 monomers and the other coarse-grained we selected was in line with previously published results.[12] A schematic of the bonding and the definition of the dihedral angle is shown in Fig. 5.1.

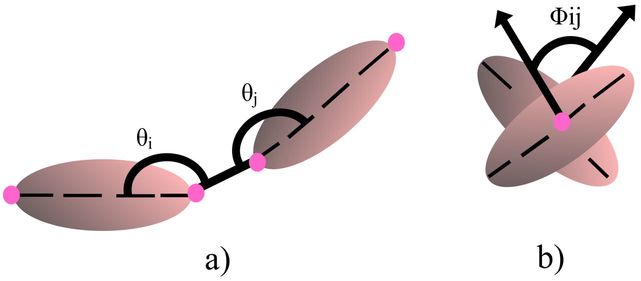


**Figure 5.1:** Schematic of anisotropic polymer used for the simulations, showing, (a) the bond angle between anisotropic monomers defined using off-centered sites and (b) the dihedral angle between adjacent monomers.

### 5.3.2 Neural network architecture

To use a machine learning approach to construct a phase space of aggregates parameterized by the polymer molecular properties and processing conditions, there has to be a systematic approach to the identification and classification of the polymer aggregates obtained from long simulations. A variational autoencoder[27] implementation is ideal for the unsupervised labeling of all configurations obtained from simulations. This variational autoencoder shown in Fig. 5.2 is constructed from an encoder network and a decoder network.[28] The encoder maps a set of inputs to a mean $\mu$ and standard deviation $\sigma$. It then samples from the standard normal distribution to create the latent space $Z$.[29,30] Using a variational autoencoder that samples from a normal distribution ensures that the latent space can be interpolated. The latent space $Z$ can then be divided into disjoint sets by resampling from a relaxed one-hot categorical distribution before reconstructing it with a decoder network. The Gumbel-softmax reparameterization trick is used to approximate an argmax function through the introduction of a neural network temperature variable.[31,32] Once determined, these disjoint sets represent the labels of different aggregates found in the training data set. During training, the neural network temperature variable is gradually reduced to anneal each configuration into a unique aggregate label. The decoder network takes the output of the encoder as an input and tries to reconstruct the input parameters of the encoder from the latent space representation. The loss function of the autoencoder is calculated as a reconstruction and regularization loss,[4] where the reconstruction error minimizes the difference between the input of the encoder and the output of the decoder and the regularization loss[33] attempts to minimize the distance between the true distribution and the distribution being sampled. This approach, where the input and the output of a feedforward neural network are the same, is considered unsupervised learning. This unsupervised learning approach reduces the prior knowledge about the polymer aggregation that is needed to find a set of the most distinct probable aggregates.

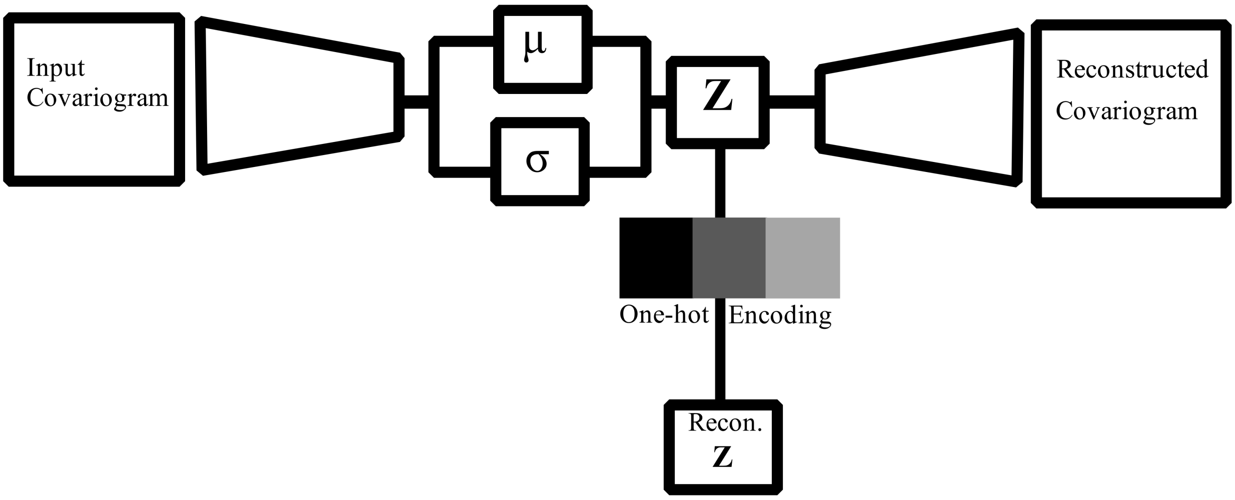


**Figure 5.2:** Schematic of autoencoder

### 5.3.3 Aggregate preprocessing

To optimize the neural network training, the polymer conformations obtained from the simulation have to be preprocessed into a representation that is invariant under translation and global rotation. Polymer configurations are first mapped to a spatial correlation matrix $\boldsymbol{M}$.[34] The $(i, j)$ element of the matrix is given by

$$M_{ij} = \boldsymbol{u}_i \cdot \boldsymbol{u}_j \tag{5.7}$$

where, $\boldsymbol{u}_i$ is the unit vector pointing from the center-of-mass of ghost atom $i$ to the center-of-mass of the $i+1$ ghost atom. This ensures that for an uncollapsed (open) polymer (Fig. 5.3c) $M_{ii} \equiv 1 \; \forall i$ and decreases exponentially along the length of the chain for all values of $M_{ij}$. Hairpin-shaped aggregates (hairpins) (Fig. 5.3g) will display a square wave pattern with a flat area close to 1 corresponding to the first arm followed by an area of rapid decay to -1 corresponding to the head and finally a flat region at -1 corresponding to the second arm going in the opposite direction. Toroidal-shaped aggregates (toroids) (Fig. 5.3e) will present with a repeating sine wave corresponding to the number of loops making up the toroid. There are no flat regions in the toroid's spatial correlation matrix because it does not possess long arms such as those seen in hairpins. A further comparison of aggregate conformation, and the corresponding spatial correlation matrix and covariogram is shown in Fig. 5.4 The 2D spatial correlation matrix is then condensed into a 1D spatial covariogram, which acts as a statistical measure of the spatial covariance as a function of distance and is calculated as[34]

$$C(h) = \frac{1}{n(h)} \sum_{j=1}^{n} \sum_{i=1}^{n} (\boldsymbol{u}_i - \boldsymbol{\mu}) \cdot (\boldsymbol{u}_j - \boldsymbol{\mu}), \tag{5.8}$$

where $h$ is the distance in space between observation $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$, $n(h)$ is the number of observations at a distance $h$, and in this case $\boldsymbol{\mu} = \vec{0}$ is the mean. $C(h)$ is a scalar function bounded between 1 and -1. The values of $h$ are chosen from the range 0 to the length of the polymer $(L)$ The cardinality of the set is fixed for all polymers and is independent of the degree of polymerization. An exponential decay corresponds to an open polymer configuration. $C(h)$ for other configurations such as multi-head rackets and toroids oscillate between 1 and -1 and the number of zero crossings corresponds to the number of heads or loops.

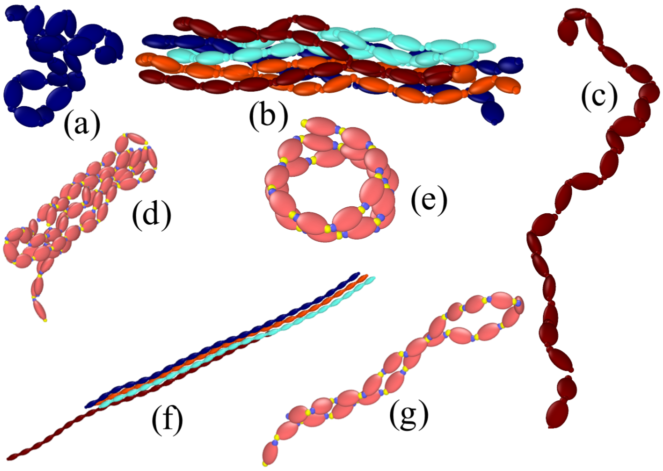


**Figure 5.3:** Typical aggregates in simulation: (a) orientationally disordered globule, (b) flexible four chain bundle, (c) open, (d) multi-head racket, (e) toroid, (f) rigid four chain bundle and (g) hairpin

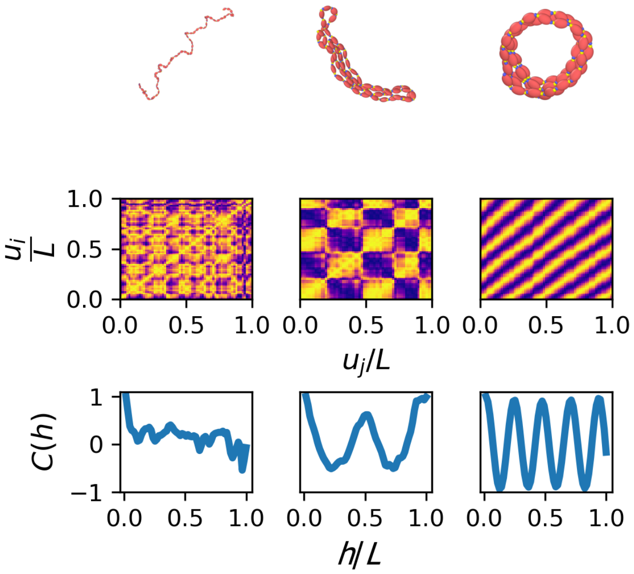


**Figure 5.4:** Typical aggregates along with the corresponding heatmap of the spatial correlation matrix and the spatial covariogram.

In the case where the number of monomers differs between polymers, the spatial covariogram will also have different length vectors. To standardize the length of the covariogram vector, the number of elements is set to 63, and where the number of monomers is less than or greater than 63, the points are interpolated using cubic spline and then 63 new points are generated along the path.

Training of the aggregate labeling neural network using a standardized data set requires the generation of data representative of possible aggregate types that would be observed in the coarse-grained polymer simulations. To generate this training data set, the list of aggregate classes considered is as follows:

1. Open is used to describe any polymer that has not collapsed into an aggregate.

2. Hairpin describes hairpin-shaped aggregates.

3. Multi-head racket describes an aggregate with more than one racket-shaped head.

4. Toroidal is used to describe all looped polymers independent of the shape or the number of loops.

A standard dataset for each of these classes of aggregates was created by manually selecting examples of the spatial covariogram associated with each of the aggregate types from the available training data and adding noise to make the training of the neural network more robust. This standardized data set ensured that all aggregates in the coarse-grained simulations were compared to and mapped to one of the possible aggregate types above. However, when the self-referential route was taken, the labeling autoencoder was trained on the spatial covariogram obtained from the simulated polymer trajectories. The training dataset obtained from molecular dynamics simulations was unbalanced due to the difference in the lifetime of various aggregates. To account for this variation in the training data, the autoencoder was trained iteratively. On the first run, a random batch of 50,000 polymer configurations was used to train the autoencoder. In each subsequent run, the trained neural network was used to evaluate the full set of available training data then the subset of data used for training was increased by 10% by adding in the polymer configurations with the largest error. The actual training data was then evaluated using the trained neural network and the bottom 1% with the smallest error was removed from the training subset. The iterative updating of the training subset was done until the average error of the training subset was equivalent to the average error of the available training data. This iterative method ensured that overrepresented configurations in the training subset were removed and rare ones were added. The autoencoder was trained on a subset of 100,000 data points from the available $4 \times 10^6$ unique polymer trajectories. The benefit of the self-referential approach over the standardized data set was that new types of aggregates can be discovered and the latent space consisted of the most probable types of aggregates. There were however some disadvantages compared to the standardized dataset. The most significant was that the aggregate classes of the latent space have to be manually labeled after the training of the neural network was completed.

The conformation of multichain aggregates was determined using the same procedure used for the single-chain aggregates. The Degree of overlap between the monomers of all pairs of polymers in a multichain aggregate was quantified using the matrix $\mathbf{\Delta}^{IJ}$, whose $(i,j)$ element is

$$\mathbf{\Delta}_{ij}^{IJ} = -\tanh\left(\frac{||\mathbf{r}_{I,i} - \mathbf{r}_{J,j}||}{\alpha\sigma}\right) + 1 \tag{5.9}$$

where $\boldsymbol{r}_{I,i}$ is the position of monomer $i$ of polymer $I$, and $\boldsymbol{r}_{J,j}$ is the position of monomer $j$ of polymer $J$, $\sigma$ is the same as Eqn. (5.3), and $\alpha$ is an integer to scale the aggregation cut-off distance.

While the covariogram describes the conformation of each polymer, the spatial matrix $\mathbf{\Delta}^{IJ}$shown in Fig. 5.5 describes the degree of overlap between polymers and highlights the position along the polymer with the highest interchain aggregation.

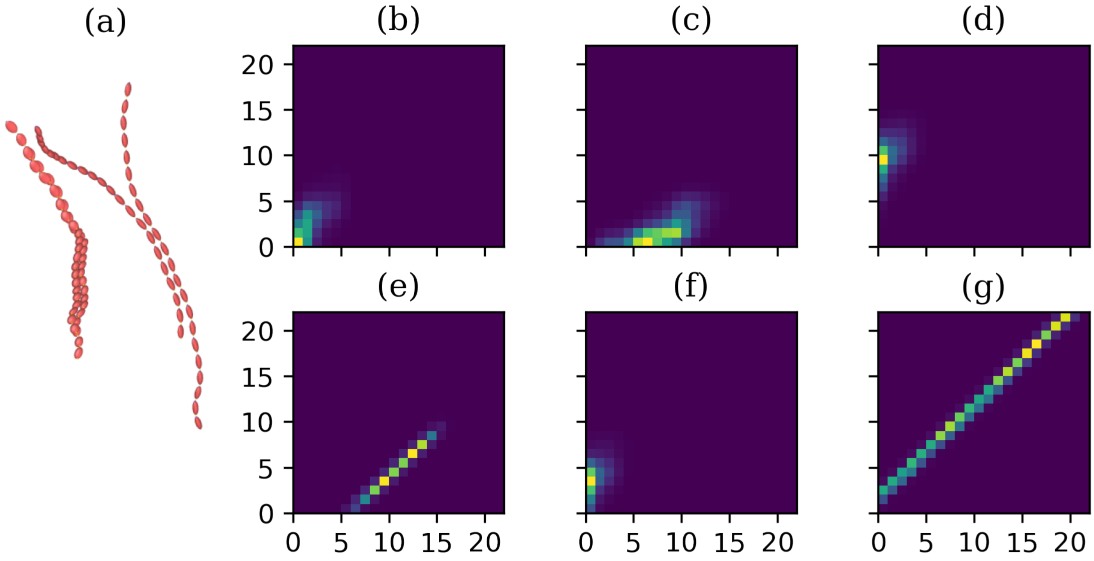


**Figure 5.5:** The six combinations of the $\mathbf{\Delta}^{IJ}$ matrix for a four polymer system (a). The partially aggregated system shows a strong alignment between two pairs of polymers in (e) and (g)

### 5.3.4 Aggregate prediction

Once all the aggregates from the molecular dynamics simulations have been labeled they could be used to predict the most probable aggregate that would occur under different conditions for a given set of polymer features. A machine learning approach allowed for the creation of high-dimensional aggregate phase diagrams. In this case, the set of polymer parameters along with the simulation condition was used as input $\mathscr{D}$ to the neural network shown in Fig. 5.6, i.e.

$$\mathscr{D} = [\tau, K_\text{A}, K_\text{D}, N, T], \tag{5.10}$$

where $K_\text{A}$ and $K_\text{D}$ are the angle and dihedral coefficient, $T$ and $N$ are the simulation temperature and degree of polymerization, and $\tau$ is the time-like variable since parallel tempering was used in the simulation, but the same analysis could be used for simulation trajectories with unbiased dynamics to predict non-equilibrium phase diagrams. All the parameters were scaled between 0 and 1 since their raw values had orders of magnitude differences.

The output of the aggregate prediction network was then compared to the labels obtained from the labeling autoencoder. The aggregate prediction neural network could explore the aggregate phase space of the polymer and visually inspect how phase boundaries change over time or with temperature and flexibility.

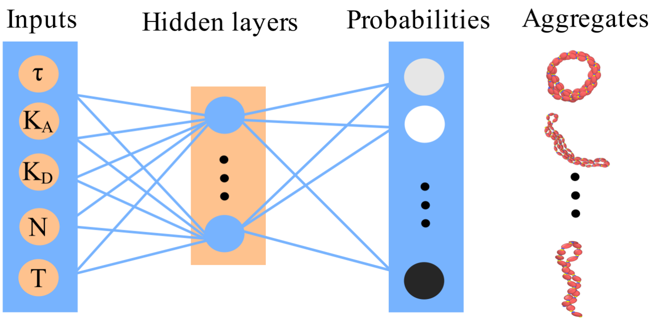


**Figure 5.6:** Schematic of classifier neural network with inputs defined in Eqn. (5.10)

### 5.3.5 Molecular Dynamics

Molecular dynamics simulations were performed using the LAMMPS package with modification to include an explicit anisotropic dihedral potential, a list of the corresponding parameters for the interaction potential can be found in the Supplementary Material. An implicit solvent model was used where the solvent was incorporated via renormalization of the intermolecular interactions and the use of the Langevin thermostat. Langevin simulations used a damping parameter of 2 and a timestep of 0.00075. Simulations were performed for chain lengths between 22 and 64 monomers in a volume of 100 and the number of chains in each simulation varied between 1 and 8. The polymer simulations were performed using parallel tempering. The temperature spacings between replicas are adjusted such that an acceptance ratio of 20–30 % is achieved for all replicas. This was used to sample a wide variety of temperatures and the complete configurational space of the polymer aggregation.

25 different simulations were done for different combinations of $K_{\mathrm{A}}$ and $K_{\mathrm{D}}$. The value of the $K_{\mathrm{A}}$ parameter was taken from the range $1 \leq K_{\mathrm{A}} \leq 5$, similarly the $K_{\mathrm{D}}$ parameter was set to a value in the range $1 \leq K_{\mathrm{D}} \leq 5$. Each simulation was done using parallel tempering with the temperature range of $0.1 \leq T \leq 1.5$ for a total of 250 different combinations of $K_{\mathrm{A}}$, $K_{\mathrm{D}}$, and $T$. Different types of polymer aggregates were observed based on the chain length and flexibility, temperature, number of chains, and the length of the simulation. The aggregates ranged from orientationally disordered globules of single chains at low temperatures and high flexibility to open rod-like multi-chain aggregates at high temperatures and low flexibility. Plots of the spatial correlation matrix and the spatial covariogram are obtained by analyzing trajectories from the simulation data. This set of known aggregate types acts as a reference key for manually assigning a name to the significant aggregate labels obtained from the encoder latent space.

The latent space of the autoencoder was set to 8 disjoint sets, to obtain eight unique aggregate labels and the neural network temperature variable was set to 2.

The temperature in the Gumbel distribution was gradually reduced by 1% each epoch until it reaches a value of 0.01. The fraction of each aggregate class was then obtained from the ratio of the number of aggregates assigned to each class to the total number of aggregates in the simulation data set.

## 5.4 Results and Discussion

Training of the autoencoding neural network produced the latent space, which can be visualized as a linear sequence of polymer aggregates parameterized by a single value, as shown in Fig. 5.7. The latent space was constructed such that aggregates with similar covariograms were grouped close to each other, ensuring smoother transitions in the phase space representation once the aggregates are given a unique aggregate label. By assigning each polymer trajectory to a unique aggregate label, the relative proportion of each aggregate in the data set could be determined. Fig. 5.8 shows the expected unbalanced dataset where the aggregate labeled A3 accounts for close to 70% of all observed aggregates.

The one-hot vector associated with the aggregate label A3 could then be passed to the decoder to find the corresponding covariogram from which the general structure of the aggregate was determined. Therefore, the decoder portion of the autoencoder must have high reconstruction fidelity. The reconstruction fidelity of the decoder can be evaluated by comparing the true and the reconstructed covariogram of data that the neural network did not use for training. The decoder portion of the autoencoder could reconstruct random selections of aggregates taken from the test set as shown in Fig. 5.9. Even with a 63:1 compression, the general shape of the covariogram was preserved with only minor deviations where the covariogram was noisy.

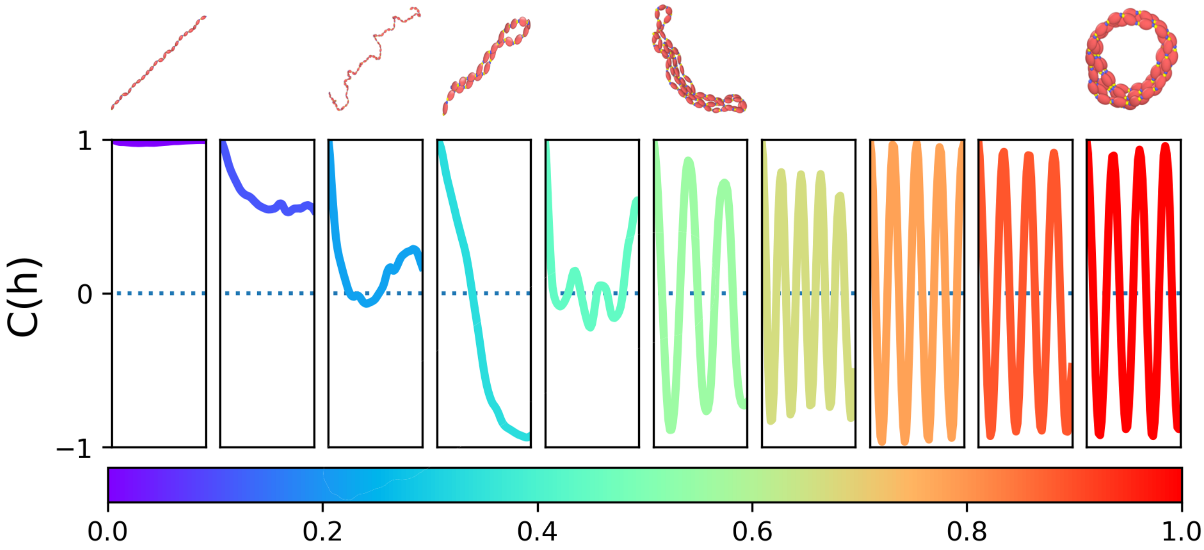


**Figure 5.7:** The reconstructed covariograms derived from a sequence of linearly spaced values in the latent space of the autoencoder. The color of each covariogram corresponds to the value of its latent space representation.

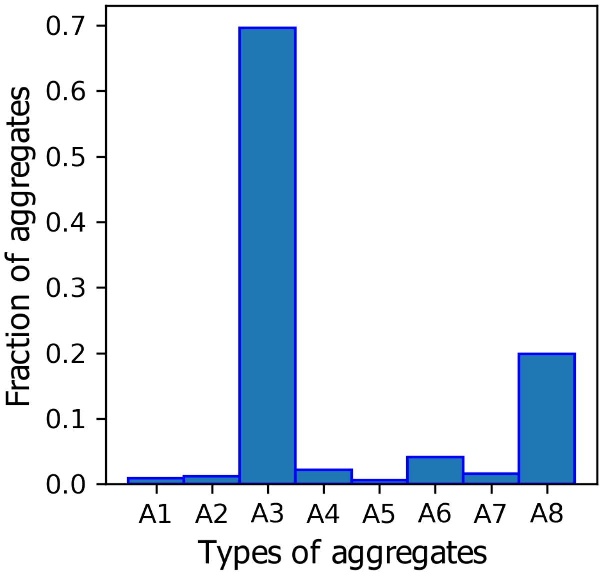


**Figure 5.8:** The relative fraction of each observed aggregate in the available training dataset.

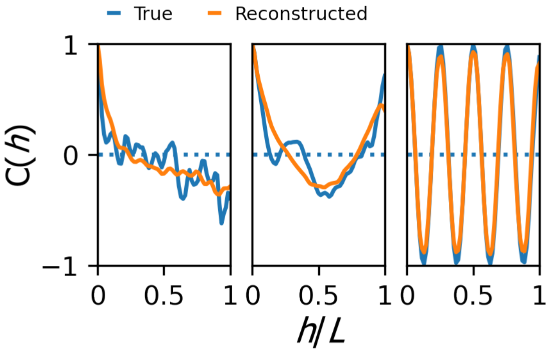


**Figure 5.9:** Examples of reconstructed covariograms corresponding to a (left) hairpin, (middle) multi-headed racket, and (right) toroid

It is expected that the conditions under which a polymer is simulated along with its intrinsic properties should determine the types of aggregates produced. When vector $\mathscr{D}$ was used as input to predict the corresponding latent variable derived from the autoencoding network, the trained classifier network could construct the expected phase space of any polymer, which lies in the range spanned by the simulated polymer trajectories. Snapshots of the neural network predicted phase space are presented in Figs. 5.10–5.12. Each slice of the high-dimensional phase space plot the aggregate latent space parameter as a function of elements of vector $\mathscr{D}$.

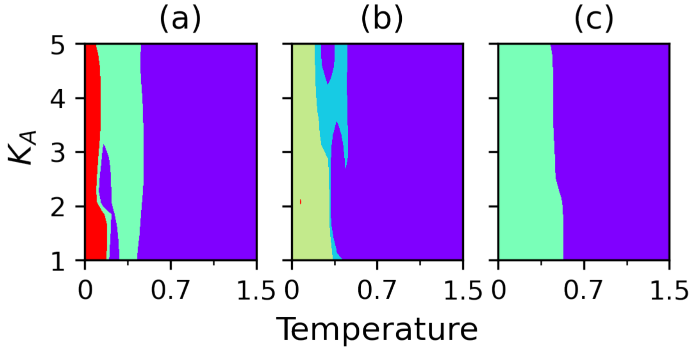


**Figure 5.10:** $K_\mathrm{A}$ versus temperature slices from the high-dimensional phase diagram of anisotropic polymer at equilibrium with $K_\mathrm{D} = 1$, for (a) 64-, (b) 44-, and (c) 22-monomer chain. The color bar and associated covariogram are shown in Fig. 5.7

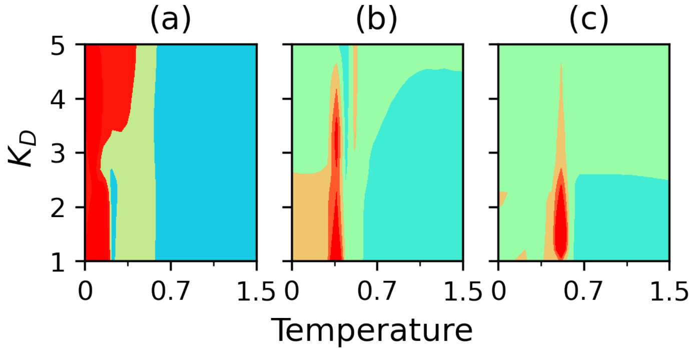


**Figure 5.11:** $K_\mathrm{D}$ versus temperature slices from the high-dimensional phase diagram of anisotropic polymer at equilibrium with $K_\mathrm{A} = 1$, for (a) 64- (b) 44- (c) 22-monomer chain. The color bar and associated covariogram are shown in Fig. 5.7

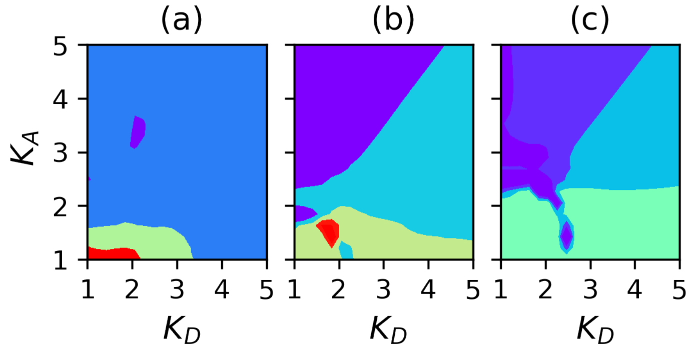


**Figure 5.12:** $K_\mathrm{A}$ versus $K_\mathrm{D}$ slices from the high-dimensional phase diagram of anisotropic polymer at equilibrium with temperature = 0.3, for (a) 64- (b) 44- (c) 22-monomer chain. The color bar and associated covariogram are shown in Fig. 5.7.

From the plots of the phase diagram, it could be determined that the open polymer dominated at high temperatures and regions where the polymer was relatively stiff. The neural network aggregate phase model also showed that flexible long-chain polymers at lower temperatures Fig. 5.10a formed a more coiled aggregate while short-chain polymers at the same temperature (Fig. 5.10c) were less likely to do so. For small values of $K_\mathrm{A}$ the anisotropic polymers are expected to aggregate at all temperatures and for all chain lengths. However, for longer chains and lower temperatures, the anisotropic polymer systems form toroids with multiple loops, as shown in Fig. 5.11. These toroidal aggregates are only expected to form for small values of $K_\mathrm{A}$ and $K_\mathrm{A}$ at low temperatures, as shown in Fig. 5.12. The length of the polymer chain also played a significant role in determining if toroidal aggregates are formed, since they are less probable for short-chained polymers shown in Fig. 5.12c The changes related to the effect of time on the long-chain semiflexible polymer are shown in Fig. 5.13, from partial collapse at small $\tau$ to the equilibrium aggregate structure at large $\tau$.

There are similarities between the neural network predicted equilibrium phase diagram shown in Fig. 5.13c and previously published results for the same set of polymer parameters and simulation conditions.[12] Both the previously published simulated phase diagram[12] and the predicted phase diagram in Fig. 5.13c show the open polymer to be the most abundant conformation at large temperature values $(T > 0.7)$ while aggregates with multiple loops were abundant at low temperatures $(T < 0.2)$. These multi-loop aggregates were independent of the value of $K_\mathrm{A}$ at low temperatures but as temperature increased, there was a transition to a single-loop aggregate at intermediate temperatures similar to previous results,[12] which showed racket-shaped aggregates as the most common at intermediate temperatures, However, the neural network predicted phase diagram in Fig. 5.13(c) does not show a distinct transition region at $1 < K_\mathrm{A} < 2$.

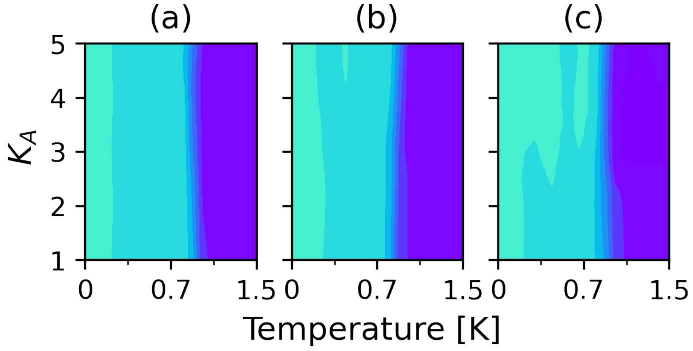


**Figure 5.13:** $K_\mathrm{A}$ versus temperature slices from the high-dimensional phase diagram of anisotropic polymer with $K_\mathrm{D} = 3$ for (a) small, (b) medium, and (c) large $\tau$. The color bar and associated covariogram are shown in Fig. 5.7

There are similarities between the phase diagrams of isotropic polymers[21] and the neural network predicted phase diagrams shown in Fig. 5.10, especially with respect to the aggregate dependence on temperature, but the transitions between aggregate phases largely happen at different temperatures and stiffness when comparing isotropic and anisotropic polymers as shown in Fig. 5.14, multi-chain aggregation of the anisotropic polymers was also similar to the aggregation of multi-chain isotropic polymers[21]. Each polymer in the pair of aggregated polymers shown in Fig. 5.14a was predicted to form a single racket-shaped aggregate at low temperatures ($T$<0.35), while Fig. 5.14b showed that the hairpins were interlocked. At higher temperatures $(T > 4)$, the neural network predicts an open configuration, and Fig. 5.14c shows that the pair are expected to completely overlap to form a rod-like aggregate. Additional phase diagrams can be found in the Supplementary Material.

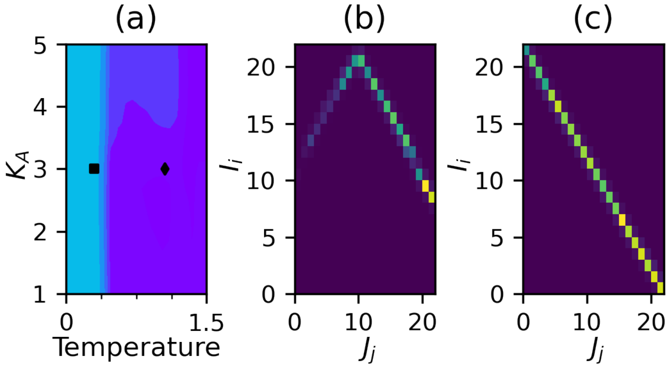


**Figure 5.14:** (a) $K_\text{A}$ versus temperature slice from the high-dimensional phase diagram of an aggregated pair of anisotropic polymer at equilibrium with $K_\text{D} = 3$ (color bar and associated covariogram are shown in Fig. 5.7). (b) The $\mathbf{\Delta}^{IJ}$ matrix for the pair of aggregated polymer with $K_\text{D} = 3$ and $T = 0.3$ and (c) the $\mathbf{\Delta}^{IJ}$ matrix for the pair of aggregated polymer with $K_\text{D} = 3$ and $T = 1.0$.

## Conclusions

An unsupervised aggregate labeling autoencoder neural network was developed to assign an aggregate type to trajectories from large simulations either by comparison to a standard set of aggregates or by a self-referential route. We further showed that this labeled data can be used alongside the polymer molecular scale parameters and the simulation conditions, to predict the most likely polymer aggregates to occur under different processing conditions, polymer flexibility, and degree of polymerization. The results confirm that there is a strong correlation between the molecular scale parameters, the processing conditions, and the equilibrium conformation of anisotropic polymer semiconductors. The neural network method was able to predict that the number of loops formed from a single chain aggregate decreases with temperature. Toroidal aggregates are also more abundant for small values of $K_\text{A}$ and $K_\text{D}$ (<2). For multi-chain aggregation, the rod-like structure was most common at equilibrium except for highly flexible polymers at low temperatures which formed interlocking hairpins. By comparing slices from the neural network constructed phase diagrams we have shown there is good agreement with previously published results using the same polymer systems. This machine learning approach, trained on coarse-grained simulations has the potential to reduce the number of atomistic simulations and experiments needed to explore the aggregate phase space when designing organic semiconductor devices. The accuracy for specific polymer systems can be further increased through top-down fine-tuning of the polymer interaction potentials and dynamics.

# References

[1] M. J. Han, D. Wei, H. S. Yun, S.-h. Lee, H. Ahn, D. M. Walba, T. J. Shin, and D. K. Yoon, NPG Asia Mater. **14**, 1 (2022).

[2] J. Lee, S. A. Park, S. U. Ryu, D. Chung, T. Park, and S. Y. Son, J. Mater. Chem. A **8**, 21455 (2020).

[3] C. Yumusak, N. S. Sariciftci, and M. Irimia-Vladu, Mater. Chem. Front. **4**, 3678 (2020).

[4] S. Wang, L. Peng, H. Sun, and W. Huang, J. Mater. Chem.C **10**, 12468 (2022).

[5] V. N. Hamanaka, E. Salsberg, F. J. Fonseca, and H. Aziz, Org. Elect. **78**, 105509 (2020).

[6] S. Allard, M. Forster, B. Souharce, H. Thiem, and U. Scherf, Angewandte Chemie Int. Ed. **47**, 4070 (2008).

[7] H. Hu, P. C. Chow, G. Zhang, T. Ma, J. Liu, G. Yang, and H. Yan, Acc. Chem. Res. **50**, 2519 (2017).

[8] S. Liu, W. M. Wang, A. L. Briseno, S. C. Mannsfeld, and Z. Bao, Adv. Mater. **21**, 1217 (2009).

[9] K. C. Dickey, J. E. Anthony, and Y.-L. Loo, Adv. Mater. **18**, 1721 (2006).

[10] M. Praprotnik, L. Delle Site, and K. Kremer, Phys. Rev. E **73**, 066701 (2006).

[11] N. E. Jackson, J. Phys. Chem. B **125**, 485 (2020).

[12] A. E. Cohen, N. E. Jackson, and J. J. De Pablo, Macromol. **54**, 3780 (2021).

[13] M. Babadi, R. Everaers, and M. Ejtehadi, J. Chem. Phys. **124**, 174708 (2006).

[14] M. Ricci, O. M. Roscioni, L. Querciagrossa, and C. Zannoni, Phys. Chem. Chem. Phys. **21**, 26195 (2019).

[15] A. F. Tillack, L. E. Johnson, B. E. Eichinger, and B. H. Robinson, J. Chem. Theory Comput. **12**, 4362 (2016).

[16] F. Goujon, N. Martzel, A. Dequidt, B. Latour, S. Garruchet, J. Devémy, R. Blaak, É. Munch, and P. Malfreyt, J. Chem. Phys. **153**, 214901 (2020).

[17] M. I. Jordan and T. M. Mitchell, Sci. **349**, 255 (2015).

[18] S. Kim, H. Wimmer, and J. Kim, in *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)* (IEEE, 2022) pp. 54–62.

[19] S. J. Wetzel, Phys. Rev.E **96**, 022140 (2017).

[20] J. Zierenberg and W. Janke, EPL **109**, 28002 (2015).

[21] J. Zierenberg, M. Marenz, and W. Janke, Poly. **8**, 333 (2016).

[22] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, Nat. **596**, 583 (2021).

[23] A. E. Cohen, N. E. Jackson, and J. J. de Pablo, Macromol. **54**, 3780 (2021).

[24] R. Berardi, C. Fava, and C. Zannoni, Chem. Phys. Lett. **297**, 8 (1998).

[25] R. Berardi, C. Fava, and C. Zannoni, Chem. Phys. Let. **236**, 462 (1995).

[26] 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, Comp. Phys. Comm. **271**, 108171 (2022).

[27] S. V. Kalinin, O. Dyck, S. Jesse, and M. Ziatdinov, Sci. Adv. **7**, eabd5084 (2021).

[28] R. Batra, H. Dai, T. D. Huan, L. Chen, C. Kim, W. R. Gutekunst, L. Song, and R. Ramprasad, Chem. Mater. **32**, 10489 (2020).

[29] C. Doersch, arXiv (2016).

[30] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020) pp. 8642–8651.

[31] E. Jang, S. Gu, and B. Poole, arXiv (2016).

[32] J. Chang, X. Zhang, Y. Guo, G. Meng, S. Xiang, and C. Pan, arXiv (2019).

[33] S. Odaibo, arXiv (2019).

[34] A. Montesi, M. Pasquali, and F. MacKintosh, Phys. Rev. E **69**, 021916 (2004).

# Chapter 6

# Conclusion

## 6.1 Summary

This work has contributed to addressing the knowledge gap that is associated with the automatic modeling of optimal coarse-grained representation of anisotropic molecules, and is applicable to a broad range of moleculecular classes, including organic polymers, nucleic acids, and liquid crystals. The development of this accurate reduced representation of molecules is especially useful to the field of organic semiconductors, for which the physical observables relevant to device properties occur over relatively large length and time scales beyond the limit of current computational tractability for atomistic systems. There has been a lot of work done in this area of research over the decades but the workflow specified in this work automates the process of mapping atomistic trajectories to a set of optimal coarse-grained coordinates with anisotropic mass distribution. The preservation of this mass distribution improves the probabilistic reconstruction of the molecule with more accurate atomic positions, bonds, and angle distributions. This back mapping is an additional beneficial feature for application to organic semiconductors being researched for application in electronic devices.

A neural network approach for the construction of coarse-grained interaction potentials has been extended to include high dimensional neural network potentials as described in Chapter 3. With the improved accuracy of machine learning potentials and the application of anisotropic sites, this work has been able to reproduce the liquid crystal behavior of one organic semiconductor using only a single-site model, which would not be possible with spherical coarse-grained sites. This project goes beyond just preserving the spatial distribution of the representation of coarse-grained molecules. Previous works such as Ref. ( 1) focused on using symmetry functions to describe the local environment of isotropic coarse-grained molecules. In this work, the symmetry functions have been modified for anisotropic particles. The most important modification is in the angular symmetry functions used to describe the orientation of a given coarse-grained particle along with the relative orientation of its neighbors. In Chapters 3 and 4, the neural network potential used for simulation extends the well-known MS-CG method to include explicit contributions from rotation-inducing torques. Obtaining an anisotropic potential from a machine-learning model required extension and modification of previous work done on the use of a prior analytical potential to capture the large repulsive energies at short separation distances.[2] In Chapters 3 and 4 an angle-dependent prior potential was developed to reduce the need to sample large amounts of data from high-energy regions. The work done here is also different from other implementations of anisotropic pair-wise potential using the AFM-CG method, since the machine learning approach can incorporate information from different state-point to produce a more accurate temperature transferable potential. While the model still suffers from some of the limitations of pure bottom-up coarse-graining, the density dependent manybody neural network potential has shown increased flexibility over traditional analytical approaches in the reproduction of the liquid crystal phases of sexithiophene using a single site model.

Unlike previously published auto-encoder algorithms, which have focused on isotropic coarse-grained sites,[3] the work done in Chapter 4 covers the automatic generation of anisotropic coarse-grained sites by extracting more information about the spatial distribution of atoms within a molecule over long time-scales. This extension of previous work on autoencoders has led to increased back-mapping fidelity. This increased accuracy in the back-mapping algorithm is especially useful for further multi-scale simulation of organic semiconductors or biological molecules. Other than the increased accuracy of the back-mapped model, the work done in Chapter 4 developed a systematic workflow to handle large molecules and polymers. The use of automatic coarse-graining has not been done on the scale of a molecule like the N2200 hexamer used for validation of the methodology. One of the major accomplishments of this work is the use of analytical bonded potential alongside a neural network potential and using gradient descent to optimize not only the neural network parameters but also the parameters of the analytical bonded potentials as well. This was accomplished by optimizing a machine learning approach to incorporate work previously done on using ghost atoms to build the bonds between anisotropic sites representing the backbone of polymers.[4]

And finally, after addressing the construction of improved coarse-grained models and interaction potentials, a machine learning workflow has been developed to automate the process of organic polymer aggregate identification in large simulation data and ultimately predict the type of polymer aggregate that is most probable under different simulation conditions and for polymers with different degree of polymerization or backbone flexibility. This is an extension of works done on isotropic and anisotropic semiflexible polymers in solution.[4,5] However, the approach presented in Chapter 5 takes a big data approach to solving the problem of understanding polymer phase space. The most notable improvement over previous work is the ability to explore and classify large amounts of aggregates with a simple workflow. the method also has predictive capabilities which are far superior to traditional linear interpolation between data points in a phase diagram. The workflow developed in Chapter 5 provides a method to explore a much higher dimensional space than was possible with previous methodologies. This high-dimensional analysis allows for an easier understanding of the role of the different factors affecting polymer aggregation. Another major benefit of the methods implemented in Chapter 5 is the use of compressed representation of polymers instead of actual polymer configurations. The resulting latent space was capable of smoothly deforming one type of aggregate into anothe aggregate which provides advantages over naive cluster analysis.

## 6.2 Future directions

The development of neural network potentials for anisotropic polymers is a natural extension of the works covered in Chapter 5. This would allow for greater integration between the methods developed in Chapters 3, 4, and 5. The work in this thesis did not attempt to match the dynamics of the coarse-grained and all-atom models of organic semiconductors. The development of machine learning techniques to address the disparity between the dynamics of the coarse-grained system and their fine-grained counterpart would improve the accuracy of the neural network potential models developed in Chapters 3 and 4. One possible approach is using generalized Langevin equation, since this has been done for spherical coarse-grained sites, but for anisotropic sites there is the added factor of rotational friction. Another future approach of this method can consider the addition of top-down information to the model to improve the representability and transferability. This approach can then be validated on a wider range of material properties beyond just forces, torques, virial, and structure. There are also opportunities to validate the model developed in Chapter 5 against real solution phase aggregation of anisotropic polymers. One possible application of the automated coarse-graining method is in high-throughput coarse-graining of a range of organic semiconductor systems to simulate large systems from which general structure-property relationships can be developed for real systems. Another future possibility is to extend the coarse-grained potentials to long-range interactions, such as electrostatics.

# References

[1] J. Behler, J. Chem. Phys. **134**, 074106 (2011).

[2] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. D. Fabritiis, F. Noé, and C. Clementi, ACS Cent. Sci. **5**, 755 (2019).

[3] W. Wang and R. Gómez-Bombarelli, npj Comput. Mater. **5**, 1 (2019).

[4] A. E. Cohen, N. E. Jackson, and J. J. De Pablo, Macromol. **54**, 3780 (2021).

[5] J. Zierenberg, M. Marenz, and W. Janke, Polymers **8**, 333 (2016).

# Statement of Authorship

| | | | |
|---|---|---|---|
| Title of Paper | Anisotropic molecular coarse-graining by force and torque matching with neural networks | | |
| Publication Status | ☐ Published<br>☐ Submitted for Publication | ☐ Accepted for Publication<br>☑ Unpublished and Unsubmitted work written in manuscript style | |
| Publication Details | | | |

## Principal Author

| | | | |
|---|---|---|---|
| Name of Principal Author (Candidate) | Marltan O. Wilson | | |
| Contribution to the Paper | Designed and trained machine learning algorithms, carried out simulations,<br>analyze and interpret the results, and compose manuscript. | | |
| Overall percentage (%) | | | |
| Certification: | This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper. | | |
| Signature | | Date | 15/12/2022 |

## Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

| | | | |
|---|---|---|---|
| Name of Co-Author | David Huang | | |
| Contribution to the Paper | Project conceptualization, research supervision, data analysis, writing (review and editing) | | |
| Signature | | Date | 09/01/2023 |

| | | | |
|---|---|---|---|
| Name of Co-Author | | | |
| Contribution to the Paper | | | |
| Signature | | Date | |

Please cut and paste additional co-author panels here as required.

# Anisotropic molecular coarse-graining by force and torque matching with neural networks

Marltan O. Wilson[1] and David M. Huang[1]
*Department of Chemistry, School of Physics, Chemistry and Earth Sciences, The University of Adelaide, Adelaide, South Australia 5005, Australia*

(*Electronic mail: david.huang@adelaide.edu.au)

(Dated: 9 January 2023)

We develop a machine-learning method for coarse-graining condensed-phase molecular systems using anisotropic particles. The method extends currently available high-dimensional neural network potentials by addressing molecular anisotropy. We demonstrate the flexibility of the method by parametrizing single-site coarse-grained models of a rigid small molecule (benzene) and a semi-flexible organic semiconductor (sexithiophene), attaining structural accuracy close to the all-atom models for both molecules. The machine-learning method of constructing the coarse-grained potential is shown to be straightforward and sufficiently robust to capture anisotropic interactions and many-body effects. The method is validated through its ability to reproduce the structural properties of the small molecule's condensed phase and the phase transitions in the semi-flexible molecule over a wide temperature range.

## I. INTRODUCTION

Machine learning is quickly becoming an invaluable tool in the search, analysis, and development of new materials.[1,2] Neural networks, in particular, have had major recent success in areas ranging from predicting the folding geometry of biological macromolecules such as proteins[3] to developing highly accurate temperature-transferable interatomic potentials.[4,5]

The latter is an important advance in the field of molecular dynamics (MD) simulations. Improvements in these machine-learning models aim to expand the length and time scale of simulations without sacrificing accuracy.[6,7] Currently used ab initio molecular dynamics simulation models are generally accurate but are computationally expensive, limiting their ability to probe long time scales.[8,9] However, neural-network potentials can produce ab initio accuracy at the computational cost of classical atomistic models.[10,11]

Even though simulations at the classical MD level are faster than ab initio MD, the speedup is still insufficient to model the long time scales needed to fully understand certain phenomena and processes such as supramolecular assembly. It is well known that explicit modeling of high-frequency motion is not critical for describing many phenomena in molecular systems. These simplifications have led to the development of molecular coarse-grained models to study large, complex materials and biological systems.[12] Parameterization of coarse-grained interaction potentials commonly takes one of two approaches: the top-down approach in which parameters are tuned to match macroscopic observables, as exemplified by the Martini model,[13] and the bottom-up approach in which interactions are derived from the properties of a fine-grained model with more degrees of freedom.[12] By following a similar bottom-up process used to apply machine learning to ab initio MD data, neural-network approaches have been extended to coarse-grained molecular models, further extending the length and time scale of simulations with atomistic accuracy.[14,15]

Neural-network potentials using isotropic coarse-grained particles have several advantages over their pair-wise additive analytical counterparts since they are constructed as many-body potentials. This many-body potential can become costly when multiple coarse-grained particles are needed to preserve the shape anisotropy. It is sometimes more accurate and computationally efficient to represent these groups of atoms as a single anisotropic coarse-grained particle such as an ellipsoid, such as in the case of large, rigid, anisotropic molecular fragments. Analytical anisotropic coarse-grained potentials such as the Gay-Berne potential[16,17] were developed to address the poor performance of spherically symmetric potentials in replicating intrinsic anisotropic interactions such as $\pi$-stacking. By modeling rigid anisotropic groups of atoms as ellipsoids, the anisotropic properties of the group are preserved in a single-site model. Shape and interaction anisotropy is especially important for the study of organic semiconductor molecules, which typically consist of highly anisotropic and rigid $\pi$-conjugated units and often form liquid-crystal phases whose morphology strongly affects their performance in devices such as solar cells, transistors, and light-emitting diodes.[18]

Unlike analytical pair-wise additive potentials such as the Gay-Berne potential, high-dimensional neural-network potentials are constructed based on the immediate neighborhood of a molecule and thus account for many-body effects as well as local density variations. Notable machine-learning implementations of inter-atomic and inter-molecular potentials include the neural-network potentials developed by Behler et al.[19] The Behler neural-network potentials are constructed from a set of symmetry functions used to represent the invariant properties of the atomic environment of each atom taken from ab initio simulations. DeepMD[10] and DeepCG[14] are two other neural-network codes constructed for atomistic and coarse-grained simulations, respectively. All of these neural-network potentials rely on an invariant representation of the atomic/molecular environment. The CGnets deep-learning approach[15] employed a prior potential to account for areas in a coarse-grained data set that may not be properly sampled due to high repulsive energies. These interactions are especially important to reproduce the local structure of the simulated material.

Machine learning has previously been applied to the parameterization of coarse-grained models with anisotropic particles,[20] but no such implementation has used a nonlinear neural-network optimization method to construct the coarse-grained potential. In this work, we address this gap in knowledge by using a neural network to construct a high-dimensional anisotropic coarse-grained potential. We parameterize the neural-network potential using a recently derived systematic and general bottom-up coarse-graining method called anisotropic force-matching coarse-graining (AFM-CG)[21] which generalizes the multi-scale coarse-graining (MS-CG) method[22] for isotropic coarse-grained particles to anisotropic particles. The method rigorously accounts for finite-temperature, many-body effects without assuming a specific functional form of the anisotropic coarse-grained potential. It yields general equations relating the forces, torques, masses, and moments of inertia of the coarse-grained particles to properties of a fine-grained (e.g. all-atom) molecular dynamics simulation based on a mapping between fine-grained and coarse-grained coordinates and momenta, and by matching the equilibrium coarse-grained phase-space distribution with the mapped distribution of the fine-grained system. The previous implementations of the AFM-CG method approximated the coarse-grained potential as a sum of pair interactions between particles.[21] Here, we extend this approach to more general many-body anisotropic interactions described by a neural network potential. We also extend the approach, which was derived for constant-volume systems in the canonical ensemble to constant-pressure systems by applying a virial-matching condition previously derived for the MS-CG method.

A general coarse-grained potential should capture any temperature-dependent phase transitions associated with either melting, annealing, or glass transition temperatures as well as the local structure and density of the material. The focus is on the development of a model for which trained parameters can be easily obtained and one capable of reproducing interaction anisotropy, temperature transferability, and many-body effects. The flexibility of the new model is demonstrated through the matching of structural and thermodynamic properties of condensed-phase systems of a small anisotropic molecule, benzene, and of a larger, more flexible organic semiconductor molecule, sexithiophene. These two molecules were chosen to determine the conditions under which coarse-grained structural inaccuracy outweighs the computational efficiency of a single-anisotropic-site model.

## II. THEORY

The key aspects of the theory that underpins the AFM-CG method and its extension to constant pressure via virial matching are summarized below. The reader is referred to Ref. 21 for a more detailed description of the AFM-CG method and the full derivation of its equations.

The positions $\boldsymbol{r}^n = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_n\}$ of the $n$ fine-grained particles are mapped onto the positions $\boldsymbol{R}^N = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_N\}$ and orientations $\boldsymbol{\Omega}^N = \{\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \ldots, \boldsymbol{\Omega}_N\}$ of the $N$ anisotropic coarse-grained particles. Each fine-grained particle $i$ is mapped to a single coarse-grained particle by defining $N$ non-intersecting subsets, $\zeta_1, \zeta_2, \ldots, \zeta_N$, of the FG particle indices such that $\zeta_I$ contains the indices of fine-grained particles mapped onto coarse-grained particle $I$. The position $\boldsymbol{R}_I$ of coarse-grained particle $I$ is defined to be equal to the center-of-mass of the group of FG particles that are mapped onto it, i.e.

$$\boldsymbol{R}_I = \frac{\sum_{i\in\zeta_I} m_i \boldsymbol{r}_i}{\sum_{i\in\zeta_I} m_i}, \tag{1}$$

where $m_i$ is the mass of FG particle $i$. The orientation

$$\boldsymbol{\Omega}_I = \begin{bmatrix} \boldsymbol{\Omega}_{I,1} \\ \boldsymbol{\Omega}_{I,2} \\ \boldsymbol{\Omega}_{I,3} \end{bmatrix} \tag{2}$$

of coarse-grained particle $I$ is specified by the rotation matrix whose components are the particle's three normalized principal axes of inertia, $\boldsymbol{\Omega}_{I,q}$ for $q = 1,2,3$. These axes are defined to be equal to the corresponding principal axes relative to the center-of-mass of the group of fine-grained particles that are mapped onto the coarse-grained particle. Thus, these axes are the normalized eigenvectors of the inertia tensor

$$\mathbb{I}_{\mathrm{FG},I} = \sum_{i\in\zeta_I} m_i(||\Delta\boldsymbol{r}_i||^2 \boldsymbol{E} - \Delta\boldsymbol{r}_i \Delta\boldsymbol{r}_i^{\mathrm{T}}), \tag{3}$$

where $\Delta\boldsymbol{r}_i = \boldsymbol{r}_i - \boldsymbol{R}_I$ is the position of fine-grained particle $i$ relative to the center-of-mass (coarse-grained particle position) and $\boldsymbol{E}$ is the $3\times3$ identity matrix. From these coordinate mappings and the relationship between generalized coordinates and momenta from Hamilton's equations,[23] mappings from the linear momenta $\boldsymbol{p}^n = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ of the fine-grained particles to the linear momenta $\boldsymbol{P}^N = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_N\}$ and angular momenta $\boldsymbol{L}^N = \{\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_N\}$ of the anisotropic coarse-grained particles can also be defined.[21] The mappings for coarse-grained particle $I$ are

$$\boldsymbol{P}_I = \frac{M_I}{\sum_{i\in\zeta_I} m_i} \sum_{i\in\zeta_I} \boldsymbol{p}_i \tag{4}$$

and

$$\boldsymbol{L}_I = \mathbb{I}_I \mathbb{I}_{\mathrm{FG},I}^{-1} \sum_{i\in\zeta_I} \Delta\boldsymbol{r}_i \times \boldsymbol{p}_i, \tag{5}$$

respectively, where $\mathbb{I}_I$ is the inertia tensor of coarse-grained particle $I$.

Given these mappings, several conditions can be derived that the coarse-grained model must satisfy for its equilibrium coarse-grained phase-space distribution to match the corresponding mapped distribution of the fine-grained system. Consistency between the configuration-space distributions gives the following matching conditions between the forces $\boldsymbol{F}_I$ and torques $\boldsymbol{\tau}_I$ on coarse-grained particle $I$ and the forces on the fine-grained particles mapped onto it:[21]

$$\boldsymbol{F}_I(\boldsymbol{R}^N, \boldsymbol{\Omega}^N) = -\frac{\partial U}{\partial R_I} = \left\langle \sum_{i\in\zeta_I} \boldsymbol{f}_i \right\rangle_{R^N,\Omega^N} \tag{6}$$

and

$$\boldsymbol{\tau}_I(\boldsymbol{R}^N,\boldsymbol{\Omega}^N) = -\sum_q \Omega_{I,q} \times \frac{\partial U}{\partial \Omega_{I,q}} = \left\langle \sum_{i\in\zeta_I} \Delta \boldsymbol{r}_i \times \boldsymbol{f}_i \right\rangle_{R^N,\Omega^N},(7)$$

where $U(R^N,\Omega^N)$ is the coarse-grained potential, $\boldsymbol{f}_i(\boldsymbol{r}^n) = -\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{r}_i}$ is the force on fine-grained particle $i$, with $\boldsymbol{u}(\boldsymbol{r}^n)$ the fine-grained potential and $\langle\cdots\rangle_{R^N,\Omega^N}$ denoting an average over fined-grained configurations mapped to coarse-grained configuration $(R^N,\Omega^N)$.

Consistency between the momentum-space distributions requires the mass $M_I$ of coarse-grained particle $I$ to be the sum of the masses of its constituent fine-grained particles, i.e.[21]

$$M_I = \sum_{i\in\zeta_I} m_i. \tag{8}$$

In addition, provided that the inertia tensor $\mathbb{I}_{\mathrm{FG},I}$ of the group of fine-grained particles mapped to this coarse-grained particle does not depend on the configuration of the other particles,[21]

$$I_{I,q}^{1/2} \exp\left(-\frac{I_{I,q}\omega_{I,q}^2}{2k_\mathrm{B}T}\right) \approx \left\langle I_{\mathrm{FG},I,q}^{1/2} \exp\left(-\frac{I_{\mathrm{FG},I,q}\omega_{I,q}^2}{2k_\mathrm{B}T}\right)\right\rangle_{\boldsymbol{R}_I,\boldsymbol{\Omega}_I}, \tag{9}$$

where $I_{I,q}$, $I_{\mathrm{FG},I,q}$, and $\omega_{I,q}$ are the components of the coarse-grained moment of inertia, fine-grained moment of inertia, and angular velocity about the $q$ axis, and $\langle\cdots\rangle_{\boldsymbol{R}_I,\boldsymbol{\Omega}_I}$ denotes an equilibrium average of fine-grained configurations consistent with the coordinate mapping of coarse-grained particle $I$. Furthermore, if the fluctuations in $I_{\mathrm{FG},I,q}$ are small compared to its mean, it can be shown that[21]

$$I_{I,q} \approx \left\langle I_{\mathrm{FG},I,q}\right\rangle_{\boldsymbol{R}_I,\boldsymbol{\Omega}_I}, \tag{10}$$

i.e. the principal moment of inertia of a coarse-grained particle about each principal axis $q$ is approximately equal to the equilibrium average of the corresponding principal moment of the fine-grained particles mapped onto it.

The AFM-CG method was derived only for the constant-volume conditions of the canonical ensemble, but is straightforwardly generalized to constant-pressure conditions by analogy with the MS-CG method for spherical coarse-grained particles in the isothermal-isobaric ensemble.[24] Thus, the force- and torque-matching conditions at constant pressure are the same as those in Eqs. (6) and (7), except that the coarse-grained forces, torques, and potential are in general functions of the coarse-grained system volume $V$ and the equilibrium average is constrained to configurations in which the fine-grained system volume $v = V$. The coarse-grained potential must also satisfy a virial-matching condition,[24]

$$\begin{aligned} W(\boldsymbol{R}^N,\boldsymbol{\Omega}^N,V) &= -\frac{\partial U}{\partial V} \\ &= \left\langle \frac{(n-N)k_\mathrm{B}T}{v} + \frac{1}{3v}\sum_{i=1}^n \boldsymbol{f}_i\cdot\boldsymbol{r}_i \right\rangle_{R^N,\Omega^N,V} \end{aligned} \tag{11}$$

In summary, for the equilibrium phase-space distribution of the coarse-grained model to match that of the fine-grained model in the isothermal-isobaric ensemble, the coarse-grained potential should satisfy Eqs. (6), (7), and (11), while the coarse-grained masses and principal moments of inertia should satisfy Eqs. (8) and (9), respectively. As shown below, using the more approximate Eq. (10) to parameterize the moments of inertia gives almost the same results as Eq. (9), even for a flexible molecule, so we have used this simpler equation for parameterization later on.

## III. METHODS

### A. Force-, torque-, and virial-matching algorithm

The analytical expression for the coarse-grain potential $U$ is not usually known. However, an approximation to the functional form can be obtained using a neural-network optimization algorithm with Eqs. (6), (7), and (11) acting as necessary constraints. In general, $U(\boldsymbol{R}^N,\boldsymbol{\Omega}^N,V)$ is a function of the particle configuration and system volume. In this work, we have assumed that $U$ does not depend explicitly on $V$, in which case[24]

$$\frac{\partial U}{\partial V} = \frac{1}{3V}\sum_{I=1}^N \frac{\partial U}{\partial \boldsymbol{R}_I}\cdot\boldsymbol{R}_I. \tag{12}$$

With this approximation, the virial-matching condition in Eq. (11) can be written, using $v = V$, as

$$-\sum_{I=1}^N \frac{\partial U}{\partial \boldsymbol{R}_I}\cdot\boldsymbol{R}_I = \left\langle 3(n-N)k_\mathrm{B}T + \sum_{i=1}^n \boldsymbol{f}_i\cdot\boldsymbol{r}_i \right\rangle_{R^N,\Omega^N,V}. \tag{13}$$

Despite this approximation, we show that the coarse-grained models parameterized later on accurately match the average density of the corresponding all-atom fine-grained system at constant pressure.

To ensure that all equivalent configurations are assigned the same position in coordinate space, a transformation was made from the set of Cartesian coordinates to a vector $\boldsymbol{D}_{IJ}$ that was invariant under translation, rotation, and permutation of any pair of coarse-grained particles $I$ and $J$,[10,25–27] which was defined in terms of the positions, $\boldsymbol{R}_I$ and $\boldsymbol{R}_J$, and orientations, $\boldsymbol{\Omega}_I$ and $\boldsymbol{\Omega}_J$, of the two particles by

$$\begin{aligned} \boldsymbol{D}_{IJ} = \{&R_{IJ}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,1}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,2}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{I,3}, \\ &\boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{R}_{IJ}\cdot\boldsymbol{\Omega}_{J,3}, \\ &\boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,1}\cdot\boldsymbol{\Omega}_{J,3}, \\ &\boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,2}\cdot\boldsymbol{\Omega}_{J,3}, \\ &\boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,3}\cdot\boldsymbol{\Omega}_{J,3}\}, \end{aligned} \tag{14}$$

where $R_{IJ} \equiv \|\boldsymbol{R}_{IJ}\|$, $\boldsymbol{R}_{IJ} \equiv \boldsymbol{R}_I - \boldsymbol{R}_J$ and $\boldsymbol{\Omega}_I$ and $\boldsymbol{\Omega}_J$ are specified by rotation matrices of the form of Eq. (2). The coordinates of each neighbor within the cut-off distance of particle $I$ were transformed to a $\boldsymbol{D}_{IJ}$ vector. All the $\boldsymbol{D}_{IJ}$ vectors for a given neighborhood were concatenated into a 2D matrix $\mathbb{D}_I$

of size $N \times \dim(\boldsymbol{D}_{IJ})$ representing a unique configurational fingerprint for coarse-grained particle $I$.

The potential function could then be written in terms of a set of neural network trainable parameters and activation functions transforming $\mathbb{D}_I$ to a potential energy value. While $\mathbb{D}_I$ is a sufficient specification of the coarse-grained coordinates to enforce relevant invariant properties of the molecular environment, it does not possess all the symmetries of the potential energy surface that it aims to fit.[25,28] For each molecular environment, it was assumed that the interactions were predominantly short-ranged such that neighbors beyond a certain cut-off distance, $R_\text{c}$, do not contribute to the potential.[19] This condition can be enforced by a cut-off function of the form

$$g_\text{c}(R_{IJ}) = \begin{cases} \frac{1}{2}\left[\cos\left(\frac{\pi R_{IJ}}{R_\text{c}}\right)+1\right], & R_{IJ} \le R_\text{c}, \\ 0, & R_{IJ} > R_\text{c}. \end{cases} \tag{15}$$

A set of these cut-off functions can enforce the radial symmetry conditions of the underlying potential energy surface by storing information about the radial distribution of neighbors according to[19]

$$G_I^1 = \sum_{J \neq I} g_\text{c}(R_{IJ}). \tag{16}$$

Continuity of the potential along angular dimensions was ensured by using a compression layer to learn a set of collective variables from vector $\boldsymbol{D}_{IJ}$ which are constrained by the well-behaved modified $G^5$ symmetry function[19] given by

$$G_I^5 = \sum_{J \neq I} \prod_{\mu=1}^{M} 2^{1-\nu}\left(1+\lambda\cos\theta_{IJ,\mu}\right)^{\nu} e^{-\eta(R_{IJ}-R_s)^2} g_\text{c}(R_{IJ}). \tag{17}$$

where $\lambda \in \{-1,1\}$ and $R_\text{s}$, $\nu$, and $\eta$ are tunable hyperparameters and $\{\cos\theta_{IJ,\mu}\}$, is the set of machine-learned collective variables with the same properties as the angular component of the underlying potential and $M$ is the total number of machine-learned angular variables. These angular symmetry functions store information about the angular-radial distribution of neighbors in the local environment of coarse-grained particle $I$ Unlike the case of spherically symmetric particles, in a local reference frame, a neighboring anisotropic particle requires a minimum of seven independent scalar variables to fully describe its position and orientation. However, previous implementations of analytical potentials, including the Gay-Berne potential,[16,17] have used fewer coordinates for the calculation of the potential and forces. Similarly, for the neural network potential, an additional compression layer was included to remove the redundant angles from the $\boldsymbol{D}_{IJ}$ vectors, since the combination of translation and rotation in 3D is parameterized by at most 7 unique coordinates. The Behler symmetry functions were enforced on the output of the compression layer, ensuring that the learned compression had the same symmetry and continuity of the underlying potential. The reduction in the dimension of $\boldsymbol{D}_{IJ}$ also decreases the amount of data that is needed to train a sufficiently accurate potential. By removing the redundant angles in $\boldsymbol{D}_{IJ}$ there is a reduced possibility of over-fitting on a small data set.

A set of these symmetry functions with tuned hyperparameters $(\lambda, \nu, \eta, R_\text{s}, R_\text{c})$ can be used to uniquely represent the structural fingerprint of the molecular environment. Symmetry functions used to represent the local environment were constructed using all possible permutations of values from a specified set of hyperparameters. Training of the neural network started with 8 symmetry functions and hyperparameters tuned to minimize the loss function, which is defined below. New symmetry functions were added to the set if they resulted in a significant reduction in the neural-network loss compared with the preceding iteration. The set of hyperparameters in the symmetry functions used in the anisotropic coarse-grained models parameterized in this work can be found in the Supplementary Material.

To further reduce the amount of data needed to train the neural network, a prior repulsive potential was defined with pairwise additive properties. This potential was used to ensure physical behavior in regions of the potential where the forces are large and thus are rarely sampled in an equilibrium molecular dynamics simulation. This prior potential only needs to satisfy two conditions: firstly, it must be repulsive at short radial separations, and, secondly, the position of its repulsive barrier must be orientationally dependent. A simple equation satisfying these conditions is

$$U_{\text{prior},I} = \sum_{J \neq I} B_1 \sigma_\text{c}\left(\mathbb{D}_I\right)^{-B_2}, \tag{18}$$

where $\sigma_\text{c}$ is a neural-network compression layer function and $B_1$ and $B_2$ are strictly positive trainable parameters. It is also possible to achieve a similar large repulsive barrier through a more advanced non-linear sampling of the molecular dynamics simulation data. $U_\text{prior}$ fits a purely repulsive potential with angular dependence to the molecular environment, while $U_\text{NN}$ fits the attractive and oscillatory corrections to the environment. The final prediction for the potential energy of the environment of coarse-grained particle $I$ is therefore the sum of the neural network potential and the prior repulsive potential,[15]

$$U_I = U_{\text{NN},I} + U_{\text{prior},I}, \tag{19}$$

and, thus, the total coarse-grained potential is

$$U = \sum_{I=1}^{N} U_I \tag{20}$$

From the matching conditions in Eqs. (6), (7), and (13), optimization of the neural-network weights and biases requires a loss function of the form

$$L = \left\langle \sum_{I=1}^{N} \left( \alpha \left| \boldsymbol{F}_{\text{FG},I} + \frac{\partial U}{\partial R_I} \right|^2 + \beta \left| \boldsymbol{\tau}_{\text{FG},I} + \sum_q \Omega_{I,q} \times \frac{\partial U}{\partial \Omega_{I,q}} \right|^2 \right) + \gamma \left| 3\,(n-N)\,k_\text{B}T + \sum_{I=1}^{N} \left( \bar{W}_{\text{FG},I} + \frac{\partial U}{\partial \boldsymbol{R}_I} \cdot \boldsymbol{R}_I \right) \right|^2 \right\rangle_{R^N, \Omega^N, V}, \tag{21}$$

where

$$\boldsymbol{F}_{\text{FG},I} \equiv \sum_{i \in \zeta_I} \boldsymbol{f}_i,\ \boldsymbol{\tau}_{\text{FG},I} \equiv \sum_{i \in \zeta_I} \Delta \boldsymbol{r}_i \times \boldsymbol{f}_i,\ \bar{W}_{\text{FG},I} \equiv \sum_{i \in \zeta_I} \boldsymbol{f}_i \cdot \boldsymbol{r}_i, \tag{22}$$

and $\alpha, \beta$, and $\gamma$ are weights. These weights specify the fraction of each loss that is used for backpropagation and were free to change with the learning rate during optimization.[14] Even though there have been significant efforts in the development of methods to fit the averaged coarse-grained forces directly,[29,30] the average total fine-grained forces subject to the constraint of matching fine-grained and coarse-grained configurations are not easily obtained. An indirect means of minimizing the loss function in Eq. (21) above is possible by replacing the constrained ensemble average with an average over instantaneous unconstrained simulation configurations,[14]

$$L_{\text{inst}} = \sum_{t=1}^{N_{\text{t}}} \Bigg[ \sum_{I=1}^{N} \Bigg( \alpha \left| \boldsymbol{F}_{\text{FG},I}(\boldsymbol{r}_t^n) + \frac{\partial U(\boldsymbol{\xi}_t)}{\partial R_I} \right|^2 + \beta \left| \boldsymbol{\tau}_{\text{FG},I}(\boldsymbol{r}_t^n) + \sum_q \Omega_{I,q}(\boldsymbol{\xi}_t)) \times \frac{\partial U(\boldsymbol{\xi}_t))}{\partial \Omega_{I,q}} \right|^2 \Bigg) + \gamma \left| 3\,(n-N)\,k_{\text{B}}T + \sum_{I=1}^{N} \left( \bar{W}_{\text{FG},I}(\boldsymbol{r}_t^n) + \frac{\partial U(\boldsymbol{\xi}_t)}{\partial \boldsymbol{R}_I} \cdot \boldsymbol{R}_I(\boldsymbol{\xi}_t) \right) \right|^2 \Bigg] \tag{23}$$

since it can be shown, for a sufficiently large dataset that comprehensively samples the equilibrium ensemble of the fine-grained system, that $L$ and $L_{\text{inst}}$ have the same global minimum. Here, $N_{\text{t}}$ is the number of simulation configurations in the dataset, $\boldsymbol{r}_t^n$ and $v_t$ are the fine-grained coordinates and system volume for configuration $t$, and $\boldsymbol{\xi}_t = (\boldsymbol{R}^N(\boldsymbol{r}_t^n), \boldsymbol{\Omega}^N(\boldsymbol{r}_t^n), V(v_t))$ is the mapped coarse-grained configuration for this fine-grained configuration. The loss function is optimized using the minibatch gradient descent as implemented in TensorFlow.

The feedforward neural network shown in Fig. 1 was then trained, where the forward propagation used matrix $\mathbb{D}_I$ as an input to predict the coarse-grained potential $U$, after which TensorFlow's computational derivative was used to calculate the outputs, namely the predicted forces, torques, and virial. In the backpropagation stage, the loss function was used to calculate the error between the true and predicted values, which was then used to update the network weights and biases. The errors between the true and predicted parameters were calculated using TensorFlow's mean squared error, and gradient descent was implemented using TensorFlow's Adam optimizer.[31] Once the error of the neural network was minimized, the neural network model was used to predict the forces, torques, and virial. However, removing the output and derivative layers gives access to the predicted potential of mean force. By optimizing the partial derivatives of the potential instead of the potential itself, by the nature of the derivative, there will be less oscillation in the potential at the edges of the data set close to the cut-off distances.

### B. LAMMPS modification and neural network implementation

The neural network was constructed in Tensorflow (version 2.3.0)[32] using the Keras (version 2.4.3) functional API[33] and saved using the Tensorflow SavedModel format. The trained neural network was implemented in LAMMPS using the Tensorflow C API and cppflow wrapper. All simulations were carried out using the LAMMPS molecular dynamics (MD) software package (version 20Nov19).[34–36] The Optimized Potentials for Liquid Simulations-All Atom (OPLS-AA) force field[37–40] was used for all all-atom simulations with a cutoff distance of 10 Å for short-ranged non-bonded interactions; long-ranged electrostatic interactions were calculated with the particle-particle particle-mesh (PPPM) method[36,41] The bonds that include hydrogen were constrained using the SHAKE algorithm.[42] Simulations were carried out in the isothermal-isobaric (NPT) ensemble at a pressure of 1 atm, with the temperature and pressure controlled by a Nosé-Hoover thermostat and barostat.[43,44]

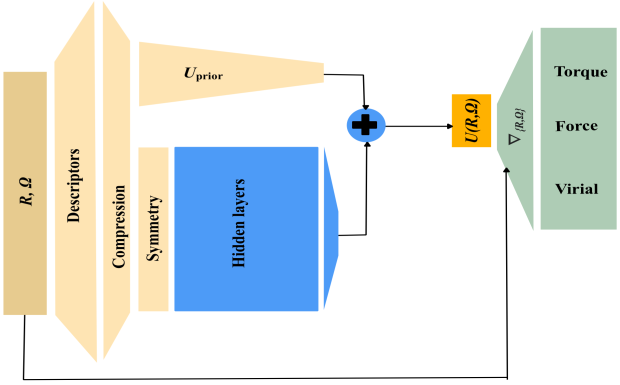
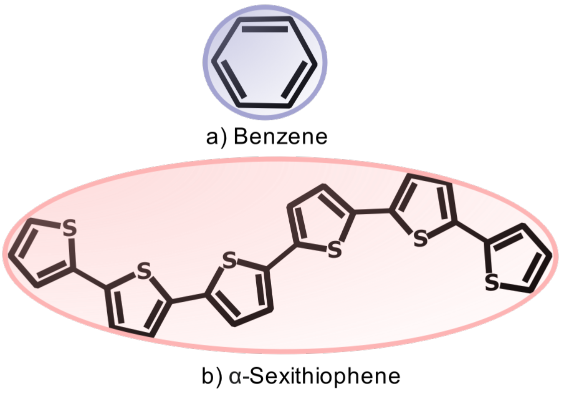


FIG. 1. Schematic of anisotropic force-matching neural network architecture.

Neural network training was carried out using data from a 25 ns all-atom simulation in which simulation configurations and forces and velocities were saved at 2 ps intervals. The simulation snapshots from the last 20 ns were shuffled and then divided into 4 groups of equal size, $\{g_0, g_1, g_2, g_3\}$. The neural network was initially trained on $g_0$ and validated on $g_3$. The validation set $g_3$ was further divided into an 8:2 ratio where the lesser was reserved as the test set. New snapshots were added from $g_1$ and $g_2$ if the mean errors of their predicted forces and torques were larger than that of the test set. The accuracy of the trained neural network was then compared to the expected accuracy determined from k-fold cross-validation.[45,46] During k-fold cross-validation, the last 20 ns of simulation data was shuffled and divided into 10 folds, $\{\psi_0, \ldots, \psi_9\}$. The model was validated on $\psi_i$ and trained on $\bigcup_{j \neq i} \psi_j$ for all $i, j \in \{0-9\}$. The loss of the iterative training method was found to be identical to the k-fold cross-validation loss.

The coarse-grained simulations were done using a modified version of the LAMMPS software where the trained neural network was introduced to calculate the forces and energies. The dimensions of the coarse-grained sites used in the simulations were derived from the inertia tensor of the all-atom model. To test the ability of the coarse-grained model to capture the properties of the all-atom model under a variety of



a) Benzene

b) α-Sexithiophene

FIG. 2. Chemical structures of (a) benzene and (b) $\alpha$-sexithiophene with coarse-grained ellipsoid superimposed on one possible configuration of each molecule.

conditions in addition to the single temperature at which the neural network was trained, the equilibrium structural properties of equivalent coarse-grained and all-atom systems were compared in simulations at several different temperatures. In all cases, the total length of the coarse-grained simulation was 25 ns long with the last 20 ns being used to calculate all structural properties. The timestep of all coarse-grained simulations was also set to 12 fs.

## IV. RESULTS AND DISCUSSION

To demonstrate the flexibility of the method we have used our neural-network model to construct coarse-grained interaction potentials for benzene, an archetypal anisotropic small molecule, and $\alpha$-sexithiophene, an organic semiconductor with significant applications in organic electronic devices[47–49] (Fig. 2). These molecules were selected to demonstrate the neural network's ability to handle anisotropic molecules of varying complexity, flexibility, and aspect ratio while still reproducing the structural and phase behavior.

The shape of a coarse-grained particle obtained from the AFM-CG method is determined by the "average" shape of the fine-grained molecule or molecular fragment that is mapped to it under the parameterization conditions. Thus, the variation of the aspect ratio of the molecule or molecular fragment with temperature in the all-atom simulations can potentially be used as a qualitative indicator of the temperature transferability of the coarse-grained model. Here, the aspect ratio of the molecule was calculated as the ratio of the length to the breadth of the molecule, where the length was defined as the longest principal axis and the breadth was defined as the sum of the remaining two semi-axes. Unlike benzene, the thiophene-thiophene torsion angles also have a temperature-dependent effect on the aspect ratio of sexithiophene.

Neural networks in general are very good at interpolation but struggle with extrapolation[50–53]. The accuracy of the model is therefore expected to decrease as the aspect ratio of the molecule deviates from that at the parameterization temperature, as well as when the density distribution is sufficiently different from the parameterization temperature. By parameterizing the systems in the liquid phase, the model can capture a wider variety of fluctuations in the density of the system and the dimensions of the molecules. The average size of a flexible molecule in the isotropic phase will be different from the size of the molecule when locked in a rigid crystal structure.[54,55] However, this temperature-dependent size difference should decrease with increased rigidity of the molecule.

### A. Benzene

Simulations consisting of 500 benzene molecules were carried out at 280, 300, 320, 330, and 350 K, and the coarse-grained neural-network model was parameterized at 300 K. The time step was 2 fs in the all-atom simulations and 12 fs in the coarse-grained simulations. The cut-off distance hyperparameter $R_c$ was 10 Å. The root mean squared validation error for the forces was 2.55 kcal mol$^{-1}$ Å$^{-1}$ and that of the torque was 4.35 kcal mol$^{-1}$. The average post-training error in the pressure calculated for the entire simulation box volume and over the entire length of the simulation at the parameterization temperature was 0.0092 atm. Benzene's average principal moments of inertia in the all-atom simulation at 300 K were used to determine the principal moments of the coarse-grained benzene model using Eqn. (10) (values given in the Supplementary Material) since fluctuations in the moments at the parameterization temperature were small.[21]

The variation of the molecular aspect ratio of the all-atom benzene model with temperature is shown in Fig. 3. The distribution of possible dimensions observed for benzene is narrow and remains fairly constant with temperature, making benzene an ideal case where molecular flexibility does not contribute significantly to the overall error of the model.[56]

Fig. 4 shows that the coarse-grained neural-network model accurately captures the liquid density of the all-atom model over a wide range of temperatures from just above the freezing point to just below the boiling point, with only slight deviations for the temperature furthest from the parameterization temperature. As shown in Fig. 5, the coarse-grained model also accurately predicts the radial distribution function (RDF) of the all-atom model over the same temperature range.

To further elucidate the accuracy of the neural network coarse-grained model, the angular-radial distribution function (ARDF) was analyzed. The ARDF is defined by

$$g(r,\theta) = \frac{\langle n(r,\theta)\rangle}{\frac{4}{3}\pi\rho\left[(r+\Delta r)^3 - r^3\right]\sin\theta\Delta\theta}, \tag{24}$$

where $\langle n(r,\theta)\rangle$ is the average number of molecules in the spherical shell within the bounds $r$ to $r+\Delta r$ of the center-of-mass of a chosen molecule and having an out-of-plane axis rotation of $\theta$ with respect to the out-of-plane axis of the chosen molecule[57], and $\rho$ is the bulk number density. Fig. 6 shows the 2D heatmap of the ARDF along with 1D slices of this function at specific angles at 300 K (the parameterization

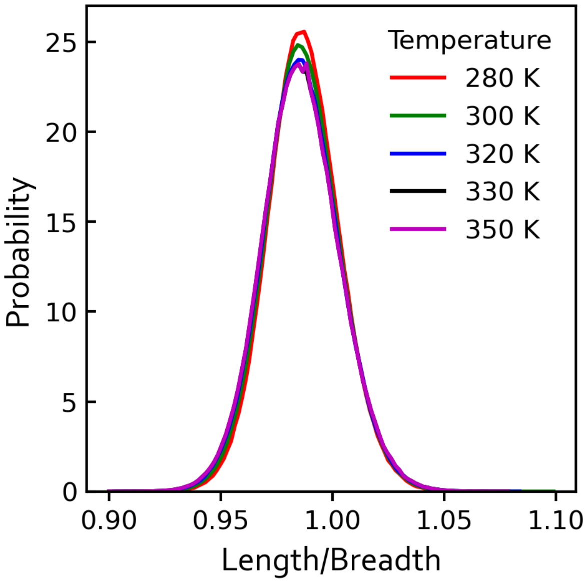


FIG. 3. Length-to-breadth ratio of the all-atom benzene model at 1 atm and various temperatures.

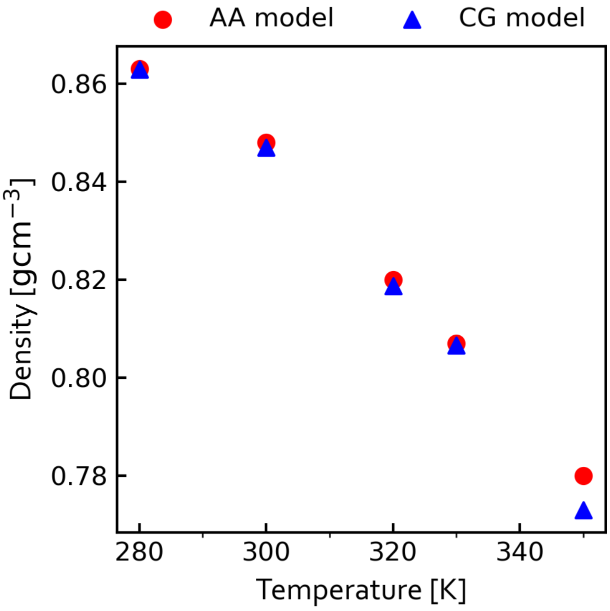


FIG. 4. Density versus temperature of the all-atom (AA) and coarse-grained (CG) benzene models at 1 atm. Error bars are smaller than the symbol

temperature) for both the all-atom and coarse-grained models. The ARDFs at the other simulated temperatures are compared in the Supplementary Material. At all simulated temperatures between 280 and 350 K, the coarse-grained model captures all the major features of the fine-grain structure of the fluid. The only difference is a slight underestimation of the peak heights by the coarse-grained model. The neural-network model is, however, able to more faithfully capture the angular radial distribution of benzene at all temperatures compared with the coarse-grained benzene model previously parameterized with the AFM-CG method using a pair potential to describe the interactions between coarse-grained particles.[21] This improvement can be attributed to the greater flexibility of the neural-network potential in describing the intermolecular interactions. The neural-network model can demonstrate temperature transferability through careful selection of the neural network hyperparameters to prevent overfitting of the local number density variations.

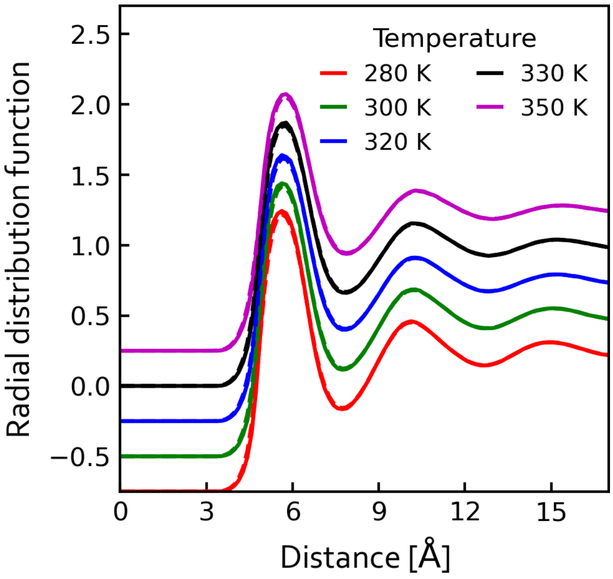


FIG. 5. Radial distribution function (RDF) of the all-atom (solid lines) and coarse-grained (dashed lines) benzene models at 1 atm and various temperatures. The RDFs have been shifted vertically for clarity.

The coarse-grained simulation of anisotropic molecules using a neural network potential is more suited for large, preferably rigid, molecules, for which a high degree of coarse-graining can be achieved. However, the model was still able to achieve a modest 20× speedup, through a combination of reduced computation time per timestep and a larger timestep, when compared to the atomistic simulations. This poor performance for a small molecule such as benzene is due to the small reduction in the number of degrees of freedom from the all-atom model to the coarse-grained model, coupled with a neural-network potential that is more computationally expensive than an analytical potential. Nevertheless, computational savings are obtained even in this suboptimal case. Simulations were carried out on a 4-core Intel i7-4790K CPU, but, further speedups could be achieved by taking advantage of the GPU-enabled version of TensorFlow.

### B. Sexithiophene

Simulations of 512 sexithiophene molecules were carried out at 570, 590, 640, and 680 K temperatures, corresponding

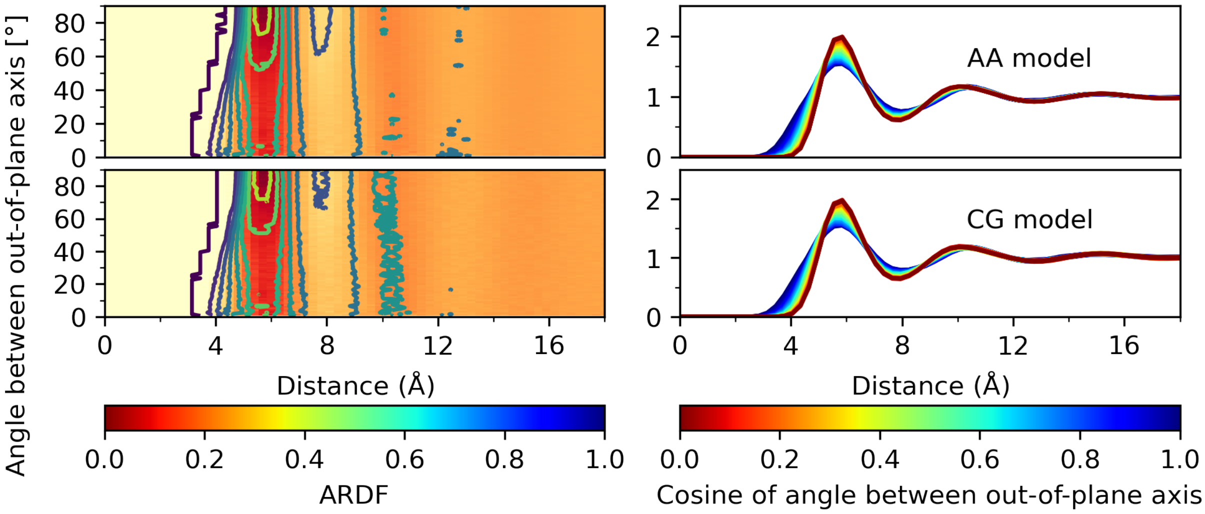


FIG. 6. Angular-radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) benzene models at 300 K and 1 atm depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on, or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90°.

to temperatures previously identified in all-atom MD simulations to correspond to crystalline (K), smectic-A (Sm-A), nematic (N), and isotropic (I) phases respectively.[58] The time step was 1 fs in the all-atom simulations and 12 fs in the CG simulations. Although we have used the OPLS-AA force field for our all-atom simulations, whereas these previous MD simulations[58] used the related AMBER force field[59–61] the structural properties of systems simulated with these two force fields (in particular the density, orientational order parameter, and radial distribution function discussed below) are very similar for the temperature range studied. The cut-off distance hyperparameter $R_c$ was set to 21 Å. The neural network model was parameterized using simulation snapshots from the isotropic phase at 680 K, where the molecular mobility was highest. The conditions of the isotropic bulk phase are advantageous in efficiently sampling the configuration space, especially rare high-energy configurations necessary for the accurate reproduction of the repulsive part of the coarse-grained potential. As shown in Fig. 7a, the distributions of the principal moments of inertia of sexithiophene in the all-atom simulation at the parameterization temperature are broad, indicating that Eqn. (10). may not be adequate for parameterizing the moments of inertia of the coarse-grained model. However, we found that using the more general Eqn. (9) to parameterize the coarse-grained moments of inertia (by fitting the distributions in Fig. 7(b–d) gave values within <1%. So we used the values from Eqn. (9) in the coarse-grained model."

The root mean squared validation error for the sexithiophene forces were 3.95 $\mathrm{kcal\,mol^{-1}\,\mathring{A}^{-1}}$ and that of the torque was 9.8 $\mathrm{kcal\,mol^{-1}}$. The sexithiophene final force and torque losses were larger than those of benzene because the model was not complex enough to account for the bending of the polymer and the rotation of the individual thiophene rings. The loss is also skewed to larger values when compared with benzene because sexithiophene is a larger molecule and so the interactions between molecules are stronger overall.

The structural properties of the coarse-grained model were compared to those of its all-atom counterpart at each of the simulated temperatures. The nonlinear change in density with respect to temperature is associated with the phase changes that occur at the simulated temperatures (Fig. 8).[58] The density of the coarse-grained system agrees well with that of the all-atom system, with minimal deviations from the fine-grained system with increasing distance from the parameterization temperature. Compared with benzene, sexithiophene has a much larger change in density between the crystalline and the isotropic phase. This difference results in less overlap between the local density variations in the crystalline phase at the lowest temperature and the training data set in the isotropic phase at the highest temperature. The sexithiophene molecule is also much more flexible than benzene, as seen in the wide distribution of the aspect ratio in the all-atom model at all the simulated temperatures shown in Fig. 9, and its dimensions change significantly with temperature over the range studied. Another limitation of representing sexithiophene as a single-site ellipsoid is the loss of thiophene–thiophene torsional information. That is, for any given position and orientation of the coarse-grained ellipsoid there are multiple different relative orientations between the thiophene groups.[62] This loss of information is significant because the anisotropic interactions of the thiophene subunits are lost, which reduces the neural network's ability to isolate which of the two short axes corresponds to the $\pi$-stacking direction.

To further confirm that the density changes were associated with transitions from crystalline through nematic and smectic to the isotropic phase, the scalar orientational order parameter $P_2$ was introduced. For a given simulation snapshot at time $t$,

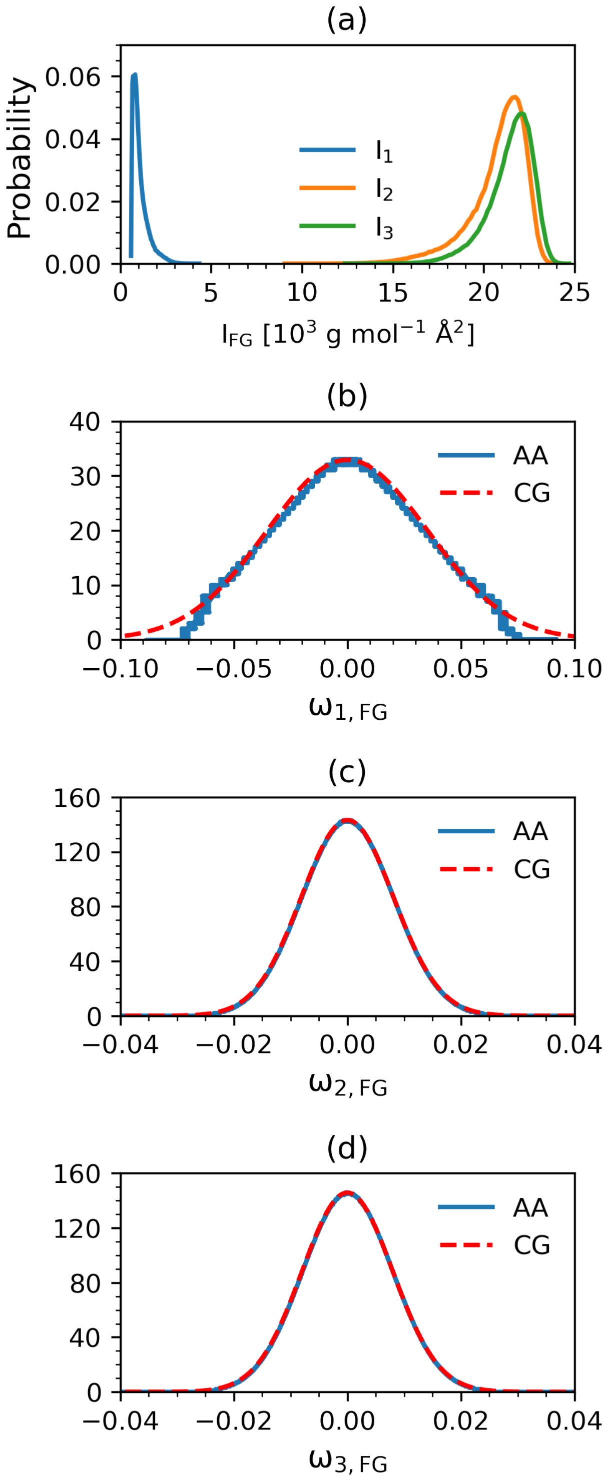


FIG. 7. (a) Principal moment of inertia distributions for all-atom (AA) sexithiophene model at 680 K and 1 atm. The corresponding angular velocity distributions of each principal axis along with the coarse-grained (CG) fit to the distribution given by Eq. (9) is shown in (b)–(d).

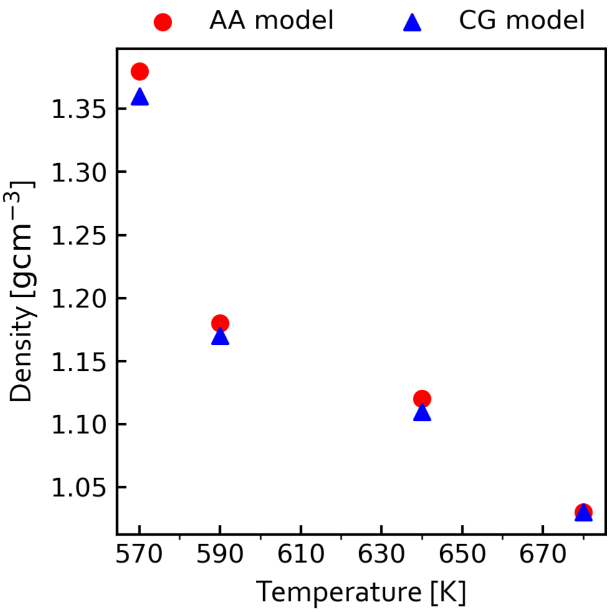


FIG. 8. Density versus temperature of the all-atom (AA) and coarse-grained (CG) sexithiophene models at 1 atm. Error bars are smaller than the symbols.

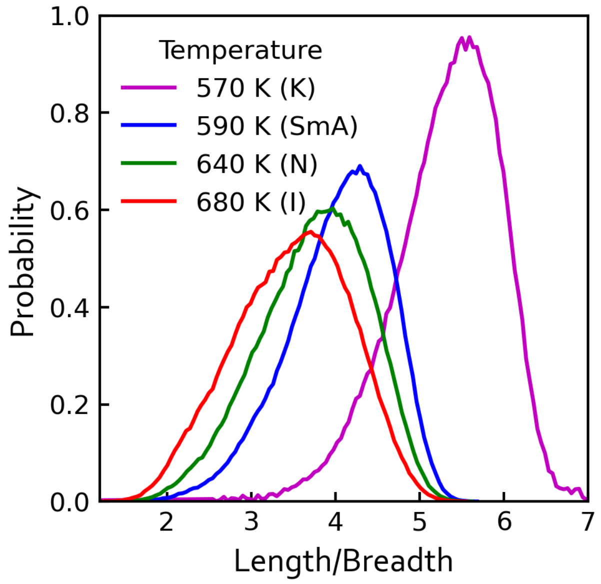


FIG. 9. Length-to-breadth ratio of all-atom sexithiophene model at 1 atm and various temperatures. The simulated phase is given in parentheses after each temperature in the legend (I = isotropic, N = nematic, SmA = smectic A, K = crystal).

$P_2$ can be found by diagonalizing the ordering matrix $\boldsymbol{Q}$,

$$\boldsymbol{Q}(t) = \frac{1}{2N}\sum_{I=1}^{N}\left[3\boldsymbol{u}_I(t)\otimes\boldsymbol{u}_I(t)-\boldsymbol{E}\right], \tag{25}$$

where $\boldsymbol{u}_I$ is the unit vector along the molecular axis and $\boldsymbol{E}$ is the identity matrix. $\langle P_2\rangle$ is the average over the largest

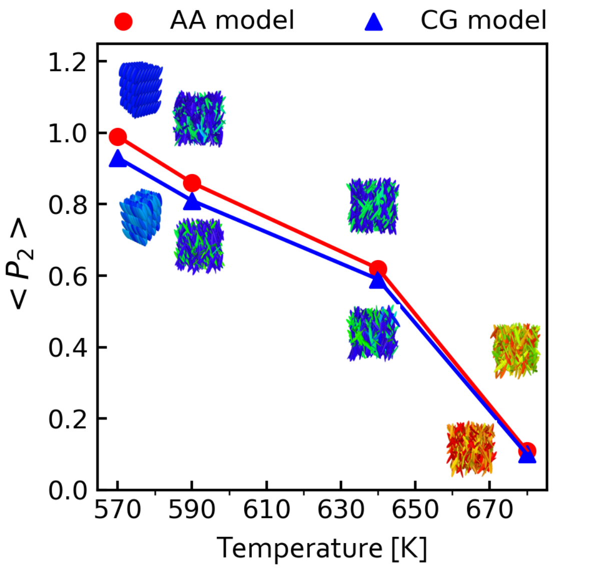


FIG. 10. Orientational order parameter versus temperature for the all-atom (AA) and coarse-grained (CG) sexithiophene models at 1 atm. Typical simulation configurations are shown at each temperature for each system (AA model above the data points and CG model below), in which the molecules have been colored according to their orientation with respect to the phase director (blue = parallel, red = perpendicular). Error bars are smaller than the symbols.

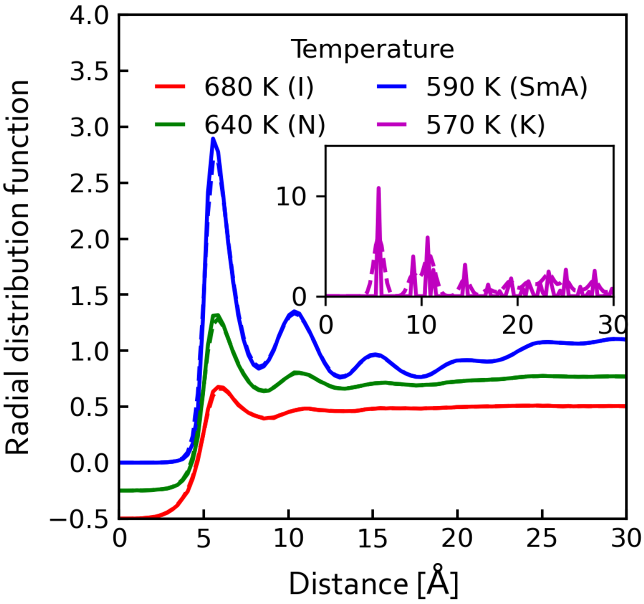


FIG. 11. Radial distribution function (RDF) of the all-atom (solid lines) and coarse-grained (dashed lines) sexithiophene models at 1 atm and various temperatures. The RDFs have been shifted vertically for clarity. The simulated phase is given in parentheses after each temperature in the legend (I = isotropic, N = nematic, SmA = smectic A, K = crystal).

eigenvalue of this matrix for all snapshots of equilibrium configurations.[58] Larger values of the scalar orientational order parameter close to one indicate an ordered crystalline structure while values close to zero correspond to an isotropic disordered phase. The coarse-grained model reproduces the orientational order parameter of the all-atom model reasonably well over the temperature range simulation, as shown in Fig. 10. The coarse-grained model underestimates the degree of orientational ordering observed in the all-atom model away from the parameterization temperature, likely because it does not capture the increasing molecular shape anisotropy that is observed in the all-atom model as the temperature decreases (Fig. 9). As expected, the largest difference occurs in the predicted crystalline phase.

The same trend is seen in the radial distribution functions shown in Fig. 11, in which the agreement between the coarse-grained and all-atom models at most temperatures is excellent, with the largest deviations for the crystalline phase. The underestimation and broadening of the peaks in the crystalline radial distribution function explain the discrepancy between the order parameter of the all-atom and coarse-grained models. The observed differences are most likely due to the effect on molecular packing of the aforementioned discrepancy in molecular shape between the two models as temperature decreases.[63] Nevertheless, even in the crystalline phase, the coarse-grained model captures the peak positions of the radial distribution function very well.

The coarse-grained model also accurately describes orientational correlations in condensed-phase sexithiophene, as illustrated by a comparison with the angular-radial distribution function of the all-atom model. At the parameterization temperature, the coarse-grained model is able to capture all major features when compared to the all-atom model (Fig. 12). The neural-network model is also able to capture the relevant features in the structure of sexithiophene's smectic liquid-crystal phase at 590 K, as shown in Fig. 13. The discrepancies in the width and height of the peaks are likely due to the differences in molecular shape away from the parameterization temperature that was mentioned earlier. The ARDFs of the two models at 640 K are compared in the Supplementary Material and show similarly good agreement.

Despite sexithiophene not strictly meeting the conditions to be coarse-grained to a single anisotropic particle due to its significant flexibility, the coarse-grained neural-network model is still able to reproduce its condensed-phase structural properties and phase behavior with remarkable accuracy. The limitation of the single-site model is only evident under conditions where the conformation of the molecule is highly temperature-dependent. One way to construct a neural network model that is independent of temperature would be to extract the training data from multiple temperatures and define the molecular dimensions as the average over the crystalline and isotropic phases. While the results for sexithiophene are substantially better than expected given its flexibility, improvements can be made to the model by considering a coarse-grained mapping consisting of more than one site.[64]

The coarse-grained simulation of sexithiophene demonstrated a speed-up of 132× compared to the all-atom simulation using the same hardware employed for the benzene sim-

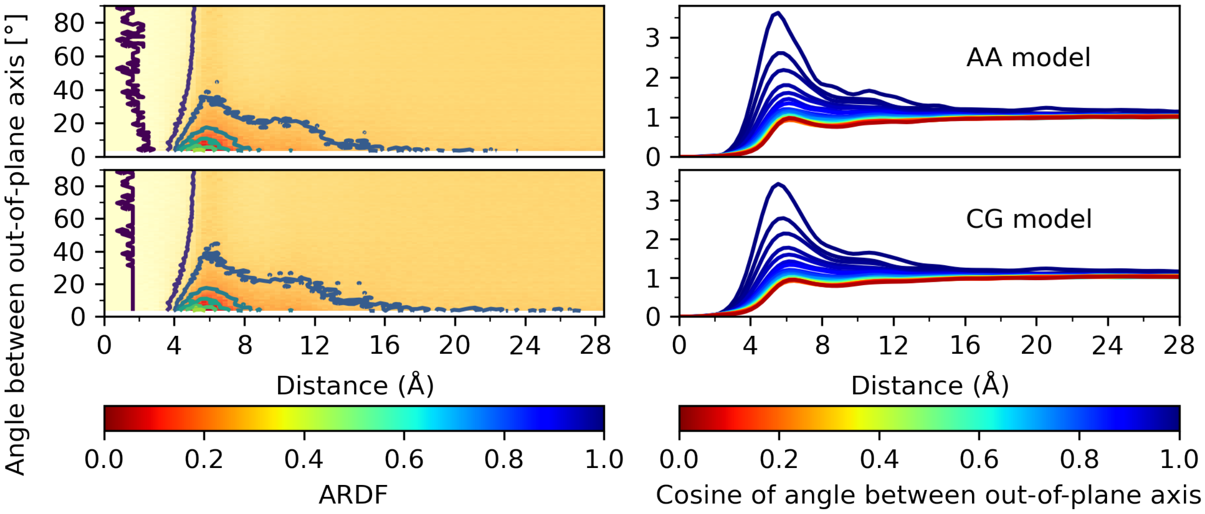


FIG. 12. Angular-radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) sexithiophene models at 680 K and 1 atm (isotropic phase) depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on, or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90 °.

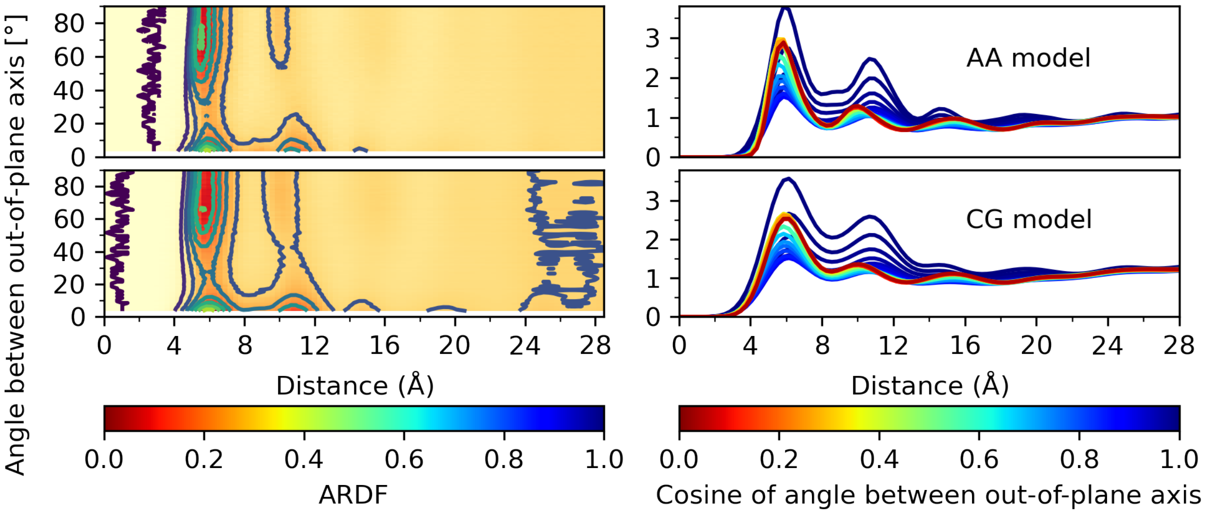


FIG. 13. Angular-radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) sexithiophene models at 590 K and 1 atm (smectic phase) depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on, or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90 °.

ulations. This speedup is primarily due to the large reduction in the number of degrees of freedom in coarse-graining this molecule.

## CONCLUSIONS

We have applied machine learning and a recently derived systematic coarse-graining method for anisotropic particles to develop a single-site anisotropic coarse-grained potential of a molecular system. The iterative training of the neural network potential is able to reproduce the forces, torques, and pressure of the fine-grained all-atom system. The final loss of the iterative training model was identical to the loss obtained from k-fold cross-validation. The CG model performs well for a rigid molecule like benzene but remarkably it also describes the phase behavior and molecular-scale structural correlations of a flexible molecule like sexithiophene with comparable accuracy, even though the aspect ratio of the molecule changes significantly over the simulated temperature range. We have demonstrated the versatility of the coarse-graining method by parameterizing models of benzene and sexithiophene at a single temperature and then studying their accuracy in capturing the structural properties of the corresponding all-atom model at different temperatures. The sexithiophene model was also used to show the ability of the model to reproduce the phase behavior of the all-atom model, with the lowest fidelity coming from the crystalline phase where the aspect ratio of the molecule has the largest deviation from the parameterization data set. A natural extension to this work would be to generalize the method to a multi-site anisotropic coarse-grained model for flexible molecules and polymers.

## ACKNOWLEDGMENTS

This work was supported by the Australian Research Council under the Discovery Projects funding scheme (DP190102100). This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government, and from the University of Adelaide's Phoenix High-Performance Computing service.

## DATA AVAILABILITY STATEMENT

The Supplementary Material provides more extensive structural comparisons between the all-atom and coarse-grained simulations, more details on the methods used for the calculations, and a list of the software needed to implement and train the neural-network potential, along with links to their GitHub repositories. Simulation files and neural network scripts can be found at https://doi.org/10.25909/21218057.v1

## REFERENCES

[1] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," Nat. **559**, 547–555 (2018).

[2] S. M. Moosavi, K. M. Jablonka, and B. Smit, "The role of machine learning in the understanding and design of materials," J. Am. Chem. Soc. **142**, 20273–20287 (2020).

[3] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, and S. Velankar, "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models," Nucl. Acids Res. **50**, D439–D444 (2021).

[4] P. Rowe, V. L. Deringer, P. Gasparotto, G. Csányi, and A. Michaelides, "An accurate and transferable machine learning potential for carbon," J. Chem. Phys. **153**, 034702 (2020).

[5] S. Stocker, J. Gasteiger, F. Becker, S. Günnemann, and J. T. Margraf, "How robust are modern graph neural network potentials in long and hot molecular dynamics simulations?" Mach. Learn.: Sci. Tech. **3**, 045010 (2022).

[6] P. Friederich, F. Häse, J. Proppe, and A. Aspuru-Guzik, "Machine-learned potentials for next-generation matter simulations," Nat. Mater. **20**, 750–761 (2021).

[7] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, "Machine learning for molecular simulation," Ann. Rev. Phys. Chem. **71**, 361–390 (2020).

[8] Z. Guo, D. Lu, Y. Yan, S. Hu, R. Liu, G. Tan, N. Sun, W. Jiang, L. Liu, Y. Chen, L. Zhang, M. Chen, H. Wang, and W. Jia, "Extending the limit of molecular dynamics with ab initio accuracy to 10 billion atoms," in *Proc. 27th ACM SIGPLAN Symp. Princ. Pract. Parallel Program.*, PPoPP '22 (Association for Computing Machinery, New York, NY, USA, 2022) pp. 205–218.

[9] D. Marx and J. Hutter, *Ab initio molecular dynamics: basic theory and advanced methods* (Cambridge University Press, 2009).

[10] H. Wang, L. Zhang, J. Han, and W. E, "DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics," Comput. Phys. Commun. **228**, 178–184 (2018).

[11] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, "Machine learning force fields," Chem. Rev. **121**, 10142–10186 (2021).

[12] J. Jin, A. J. Pak, A. E. Durumeric, T. D. Loose, and G. A. Voth, "Bottom-up coarse-graining: Principles and perspectives," J. Chem. Theory Comput. **18**, 5759–5791 (2022).

[13] S. J. Marrink and D. P. Tieleman, "Perspective on the martini model," Chem. Soc. Rev. **42**, 6801–6822 (2013).

[14] L. Zhang, J. Han, H. Wang, R. Car, and W. E. Weinan, "DeePCG: Constructing coarse-grained models via deep neural networks," J. Chem. Phys. **149**, 034101 (2018).

[15] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. D. Fabritiis, F. Noé, and C. Clementi, "Machine learning of coarse-grained molecular dynamics force fields," ACS Cent. Sci. **5**, 755–767 (2019).

[16] R. Berardi, C. Fava, and C. Zannoni, "A generalized Gay-Berne intermolecular potential for biaxial particles," Chem. Phys. Lett. **236**, 462–468 (1995).

[17] J. G. Gay and B. J. Berne, "Modification of the overlap potential to mimic a linear site–site potential," J. Chem. Phys. **74**, 3316–3319 (1981).

[18] B. J. Boehm, H. T. Nguyen, and D. M. Huang, "The interplay of interfaces, supramolecular assembly, and electronics in organic semiconductors," J. Phys.: Cond. Matter **31**, 423001 (2019).

[19] J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," J. Chem. Phys. **134**, 074106 (2011).

[20] G. Campos-Villalobos, G. Giunta, S. Marín-Aguilar, and M. Dijkstra, "Machine-learning effective many-body potentials for anisotropic particles using orientation-dependent symmetry functions," J. Chem. Phys. **157**, 024902 (2022).

[21] H. T. L. Nguyen and D. M. Huang, "Systematic bottom-up molecular coarse-graining via force and torque matching using anisotropic particles," J. Chem. Phys. **156**, 184118 (2022).

[22] W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, "The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models," J. Chem. Phys. **128**, 244114 (2008).

[23] H. Goldstein, *Classical Mechanics* (Addison-Wesley San Francisco, 2002).

[24] S. A. Colgate, "Non-lte astrophysics and the origin of high energy cosmic rays," in *AIP Conference Proceedings*, Vol. 96 (American Institute of Physics, 1983) pp. 306–312.

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

[26] J. Han, L. Zhang, R. Car, and W. E, "Deep potential: A general representation of a many-body potential energy surface," Commun. Comput. Phys. **23**, 629–639 (2018).

[27] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *2019 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR* (IEEE, 2019) pp. 5738–5746.

[28] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, "wACSF—Weighted atom-centered symmetry functions as descriptors in machine learning potentials," J. Chem. Phys. **148**, 241709 (2018).

[29] G. Ciccotti, R. Kapral, and E. Vanden-Eijnden, "Blue moon sampling, vectorial reaction coordinates, and unbiased constrained dynamics," ChemPhysChem **6**, 1809–1814 (2005).

[30] J. B. Abrams and M. E. Tuckerman, "Efficient and direct generation of multidimensional free energy surfaces via adiabatic dynamics without coordinate transformations," J. Phys. Chem. B **112**, 15742–15757 (2008).

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv , 1412.6980 (2014).

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symp. Oper. Syst. Des. Implement. OSDI 16* (2016) pp. 265–283.

[33] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras (2015).

[34] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," J. Comput. Phys. **117**, 1–19 (1995).

[35] W. M. Brown, P. Wang, S. J. Plimpton, and A. N. Tharrington, "Implementing molecular dynamics on hybrid high performance computers – short range forces," Comput. Phys. Commun. **182**, 898–911 (2011).

[36] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington, "Implementing molecular dynamics on hybrid high performance computers – Particle–particle particle–mesh," Comput. Phys. Commun. **183**, 449–459 (2012).

[37] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, "Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids," J. Am. Chem. Soc. **118**, 11225–11236 (1996).

[38] W. L. Jorgensen and N. A. McDonald, "Development of an all-atom force field for heterocycles. Properties of liquid pyridine and diazenes," J. Mol. Struct. THEOCHEM **424**, 145–155 (1998).

[39] R. C. Rizzo and W. L. Jorgensen, "OPLS all-atom model for amines: Resolution of the amine hydration problem," J. Am. Chem. Soc. **121**, 4827–4836 (1999).

[40] M. L. Price, D. Ostrovsky, and W. L. Jorgensen, "Gas-phase and liquid-state properties of esters, nitriles, and nitro compounds with the OPLS-AA force field," J. Comput. Chem. **22**, 1340–1352 (2001).

[41] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (CRC Press, 1998).

[42] J.-P. Ryckaert, G. Ciccotti, and H. J. Berendsen, "Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes," J. Comput. Phys. **23**, 327–341 (1977).

[43] W. G. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," Phys. Rev. A **31**, 1695 (1985).

[44] S. Nosé, "A molecular dynamics method for simulations in the canonical ensemble," Mol. Phys. **52**, 255–268 (1984).

[45] B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?" Comput. Statist. **36**, 2009–2031 (2021).

[46] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," Adv. Neural Inf. Process. Syst. **16** (2003).

[47] H. Katz, "Organic molecular solids as thin film transistorsemiconductors," J. Mater. Chem. **7**, 369–376 (1997).

[48] D. Fichou, "Structural order in conjugated oligothiophenes and its implications on opto-electronic devices," J. Mater. Chem. **10**, 571–588 (2000).

[49] Y. Dong, V. C. Nikolis, F. Talnack, Y.-C. Chin, J. Benduhn, G. Londi, J. Kublitski, X. Zheng, S. C. Mannsfeld, D. Spoltore, *et al.*, "Orientation dependent molecular electrostatics drives efficient charge generation in homojunction organic solar cells," Nat. commun. **11**, 1–9 (2020).

[50] P. J. Haley and D. Soloway, "Extrapolation limitations of multilayer feedforward neural networks," in *Proc. 1992 IJCNN Int. Jt. Conf. Neural Netw.*, Vol. 4 (IEEE, 1992) pp. 25–30.

[51] G. S. Na, S. Jang, and H. Chang, "Nonlinearity encoding to improve extrapolation capabilities for unobserved physical states," Phys. Chem. Chem. Phys. **24**, 1300–1304 (2022).

[52] Y. Ding, A. Pervaiz, M. Carbin, and H. Hoffmann, "Generalizable and interpretable learning for configuration extrapolation," in *Proc. 29th ACM Jt. Meet. Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.* (2021) pp. 728–740.

[53] J. P. Rigol, C. H. Jarvis, and N. Stuart, "Artificial neural networks as a tool for spatial interpolation," Int. J. Geogr. Inf. Sci. **15**, 323–343 (2001).

[54] M. Mueller, J. Zierenberg, M. Marenz, P. Schierz, and W. Janke, "Probing the effect of density on the aggregation temperature of semi-flexible polymers in spherical confinement," Phys. Procedia **68**, 95–99 (2015).

[55] D. Seaton, S. Mitchell, and D. Landau, "Monte Carlo simulations of a semi-flexible polymer chain: a first glance," Braz. J. Phys. **36**, 623–626 (2006).

[56] A. V. Sinitskiy, M. G. Saunders, and G. A. Voth, "Optimal number of coarse-grained sites in different components of large biomolecular complexes," J. Phys. Chem. B **116**, 8363–8374 (2012).

[57] M. Falkowska, D. T. Bowron, H. G. Manyar, C. Hardacre, and T. G. Youngs, "Neutron scattering of aromatic and aliphatic liquids," ChemPhysChem **17**, 2043–2055 (2016).

[58] A. Pizzirusso, M. Savini, L. Muccioli, and C. Zannoni, "An atomistic simulation of the liquid-crystalline phases of sexithiophene," J. Mater. Chem. **21**, 125–133 (2011).

[59] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner, "A new force field for molecular mechanical simulation of nucleic acids and proteins," J. Am. Chem. Soc. **106**, 765–784 (1984).

[60] S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, "An all atom force field for simulations of proteins and nucleic acids," J. Comput. Chem. **7**, 230–252 (1986).

[61] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," J. Am. Chem. Soc. **117**, 5179–5197 (1995).

[62] F. D. Tsourtou, S. D. Peroukidis, L. D. Peristeras, and V. G. Mavrantzas, "Monte Carlo algorithm based on internal bridging moves for the atomistic simulation of thiophene oligomers and polymers," Macromol. **51**, 8406–8423 (2018).

[63] W. Xia, N. K. Hansoge, W.-S. Xu, F. R. Phelan Jr, S. Keten, and J. F. Douglas, "Energy renormalization for coarse-graining polymers having different segmental structures," Sci. Adv. **5**, eaav4683 (2019).

[64] G. D'Adamo, R. Menichetti, A. Pelissetto, and C. Pierleoni, "Coarse-graining polymer solutions: A critical appraisal of single-and multi-site models," Eur. Phys. J. Special Topics **224**, 2239–2267 (2015).

# Supplementary Material:
# Anisotropic molecular coarse-graining by force and torque matching with neural networks

Marltan O. Wilson and David M. Huang

*Department of Chemistry, School of Physics, Chemistry and Earth Sciences,*

*The University of Adelaide, Adelaide, South Australia 5005, Australia*

## CONTENTS

## SI. BENZENE NETWORK PARAMETERS

The cut-off distance $R_\mathrm{c}$ for the benzene neural network was set at 10 Å for the $G^1$ type symmetry function. For the $G^5$ angular symmetry function, $\lambda$ had values of -1 and 1. $\nu$ has values of $2^n$, where $n \in \mathbb{Z}$. Hyperparameters $\alpha$, $\beta$, and $\gamma$ in the loss function were adjusted to improve the speed of convergence, but this did not usually affect the global minimum of the optimization when the number of training epochs was large.

TABLE S1: Hyperparameters for benzene.

| hyperparameter | value | units |
|---|---|---|
| $\lambda$ | [-1.0, 1.0] | |
| $\eta$ | [2.0, 1.0] | $\mathrm{\AA}^{-2}$ |
| $\nu$ | [2.0, 4.0, 8.0, 16.0, 32.0, 64.0] | |
| $R_\mathrm{s}$ | [3.0,3.7, 4.3, 5.0, 5.7, 6.3, 7.0, 7.7, 8.3, 9.0] | Å |
| $R_\mathrm{c}$ | [10.0] | Å |

## SII. ADDITIONAL BENZENE STRUCTURAL DISTRIBUTIONS

TABLE S2: The optimal coarse-grained principal moments of inertia $I_q$ for $q = 1, 2, 3$, calculated using Eqs. (10) of the main paper.

| principal axis $q$ | $I_q$ ($\mathrm{g\,mol^{-1}\,\mathring{A}^{-2}}$) |
|---|---|
| 1 | 88.1 |
| 2 | 92.2 |
| 3 | 180.1 |

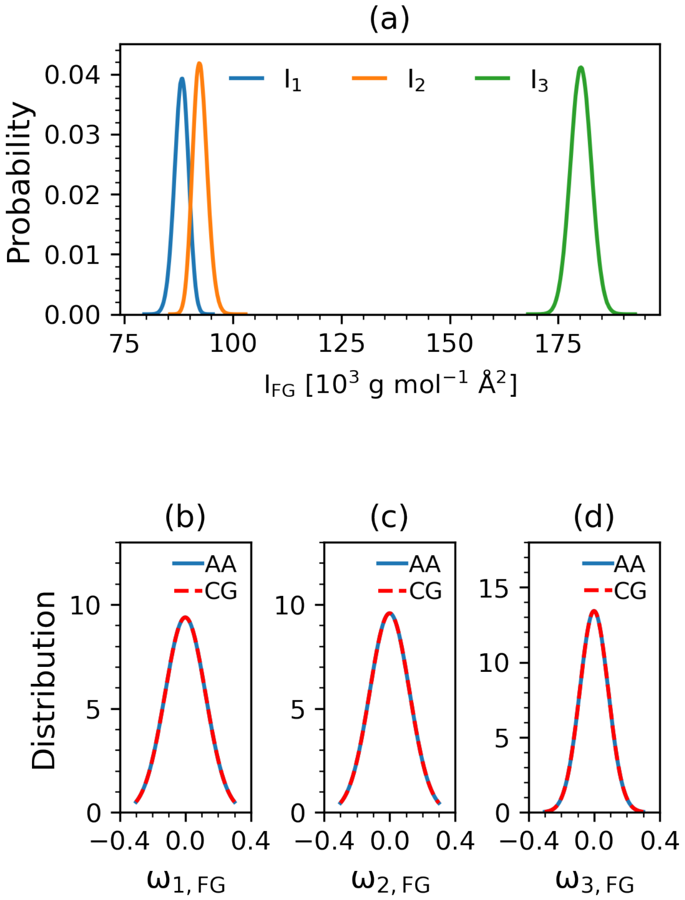


FIG. S1: (a) Principal moment of inertia distributions for the all-atom (AA) benzene model at 300 K and 1 atm. The corresponding angular velocity distributions of each principal axis along with the coarse-grained (CG) fit to the distribution given by $I_q^{1/2}\exp(-\frac{I_q\omega_q^2}{2k_BT})$ is shown in (b)–(d).

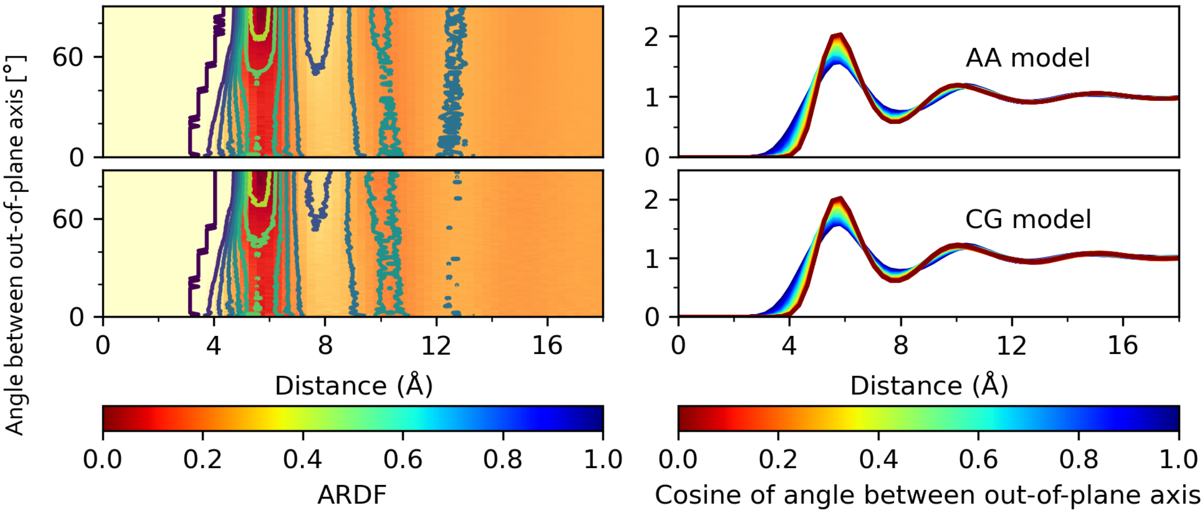


FIG. S2: Angular–radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) benzene models at 280 K and 1 atm depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90°.

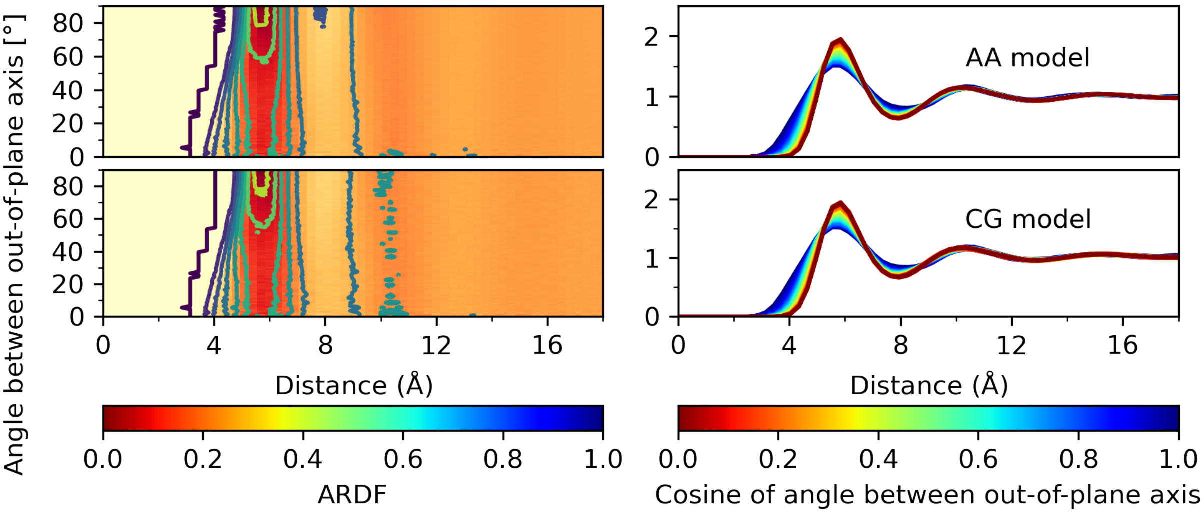


FIG. S3: Angular–radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) benzene models at 320 K and 1 atm depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90°.

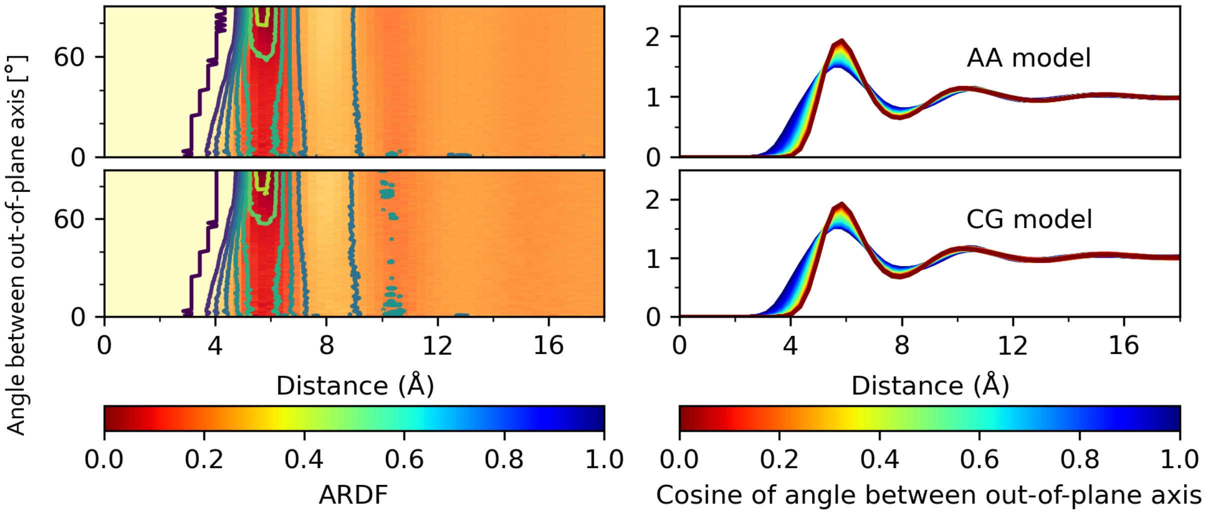


FIG. S4: Angular–radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) benzene models at 330 K and 1 atm depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90°.

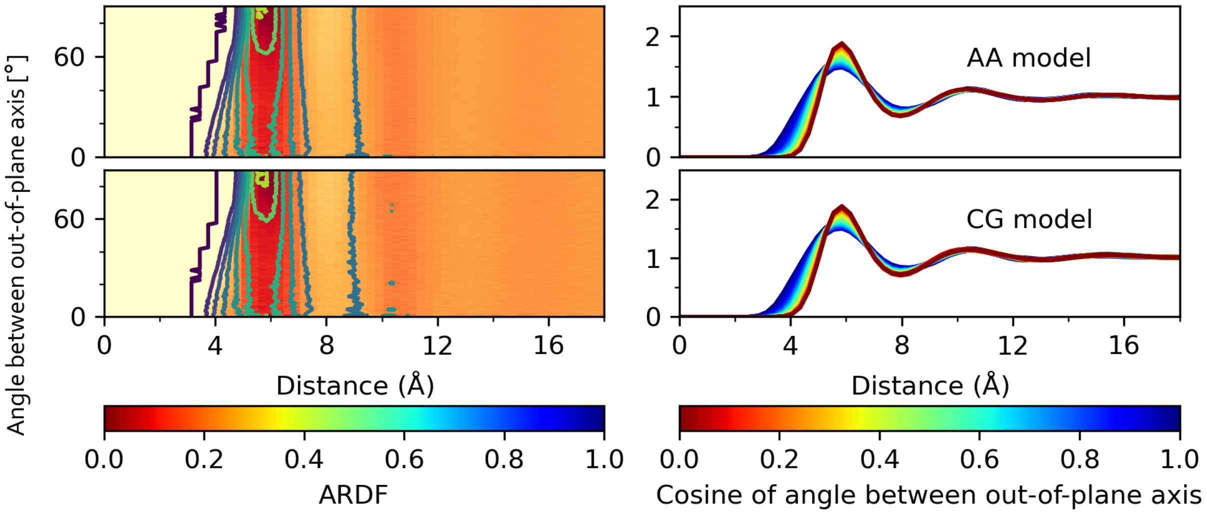


FIG. S5: Angular–radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) benzene models at 350 K and 1 atm depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90°.

## SIII. SEXITHIOPHENE NETWORK PARAMETERS

The cut-off distance $R_\mathrm{c}$ for the sexithiophene neural network was set to 21 Å for the $G^1$ type symmetry function. For the $G^5$ angular symmetry function $\lambda$ had values of -1 and 1, $\nu$ has values of $2^n$ where $n \in \mathbb{Z}$. Hyperparameters $\alpha$, $\beta$, and $\gamma$ in the loss function were adjusted to improve the speed of convergence but did not usually affect the global minimum of the optimization when the number of training epochs was large.

TABLE S3: Hyperparameters for sexithiophene

| hyperparameter | value | units |
|---|---|---|
| $\lambda$ | [-1.0, 1.0] | |
| $\eta$ | [2.0, 1.0] | $\text{Å}^{-2}$ |
| $\nu$ | [2.0, 4.0, 8.0, 16.0, 32.0, 64.0] | |
| $R_\mathrm{s}$ | [0.5, 2.7, 5.0, 7.3, 9.6, 11.8, 14.2, 16.4, 18.7, 21.0] | Å |
| $R_\mathrm{c}$ | [ 21.0] | Å |

## SIV. ADDITIONAL SEXITHIOPHENE STRUCTURAL DISTRIBUTIONS

TABLE S4: Optimal coarse-grained principal moments of inertia $I_q$ for $q = 1, 2, 3$, calculated using Eqs. (9) and (10) of the main paper and the percentage difference between these values.

| principal axis $q$ | $I_q$ (Eq. (9)) ($\mathrm{g\,mol^{-1}\,Å^{-2}}$) | $I_q$ (Eq. (10)) ($\mathrm{g\,mol^{-1}\,Å^{-2}}$) | % difference |
|---|---|---|---|
| 1 | 1083.0 | 1080.8 | 0.2 |
| 2 | 20 543.1 | 20 712.5 | 0.8 |
| 3 | 21 280.8 | 21 395.8 | 0.5 |

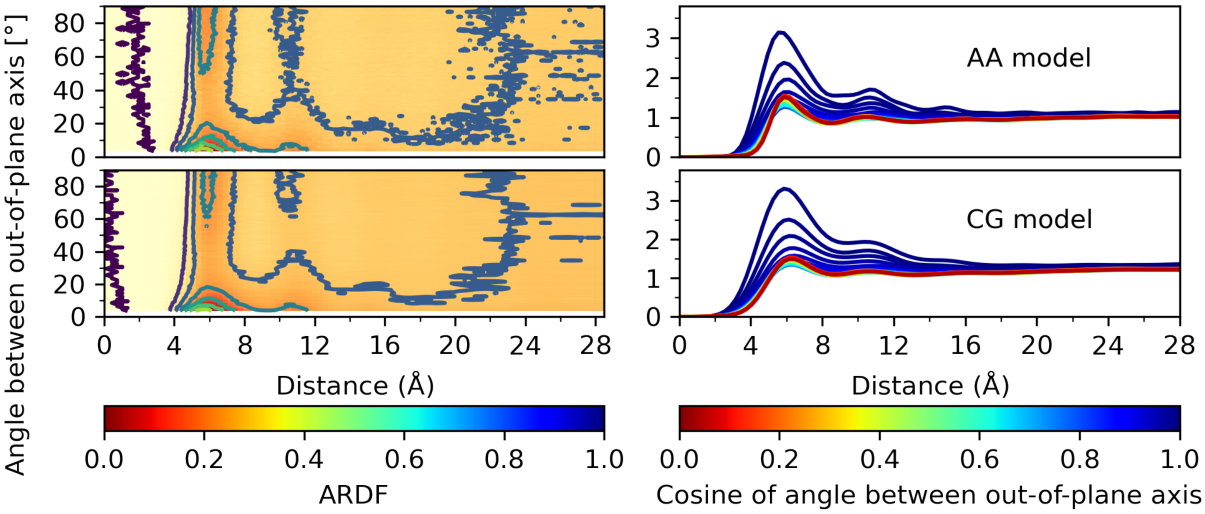


FIG. S6: Angular–radial distribution function (ARDF) of the all-atom (AA) (top) and coarse-grained (CG) (bottom) sexithiophene models at 640 K and 1 atm depicted as a heat map (left) and 1D slices at constant angle (right). Face-on, edge-on or parallel displaced configurations occur when the angle is 0°, while T-shape and Y-shape configurations occur at 90°.

## SV. TENSORFLOW AND LAMMPS IMPLEMENTATION REQUIREMENTS

The following list of software is needed to train and use the neural network model in coarse–grained simulations.

1. Tensorflow C API [https://github.com/tensorflow/tensorflow/blob/master/tensorflow/c/c_api.h]

2. Cpp Flow [https://github.com/serizba/cppflow]

3. Tensorflow Python [https://github.com/tensorflow/tensorflow]

4. Keras [https://github.com/keras-team/keras]

5. LAMMPS [https://github.com/lammps/lammps]

The training and testing of the neural network potential was done with TensorFlow in Python using the Keras functional API. The force and torque calculations were obtained through TensorFlow's Gradient Tape feature, which provides computational derivatives with respect to the network parameters. The tanh activation function was used for all standard neural network layers except the output layer since the tanh activation produced a smooth differentiable potential energy surface. The mean squared error was used when calculating the loss for the forces, torques, and virials. The Adam optimizer[1] was used as the gradient descent algorithm since it was able to reach the global minimum without manually updating the learning rate during training. The machine-learning potential was deployed with the TensorFlow C API and Cpp Flow wrapper. Cpp Flow allows the TensorFlow C model to be accessed directly as a force and torque calculator in a LAMMPS pair-style function.

## REFERENCES

[1] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv , 1412.6980 (2014).

# Statement of Authorship

| | |
|---|---|
| Title of Paper | Automated anisotropic coarse-graining of polymers using variational autoencoders |
| Publication Status | □ Published<br>□ Accepted for Publication<br>□ Submitted for Publication<br>☑ Unpublished and Unsubmitted w ork w ritten in manuscript style |
| Publication Details | |

## Principal Author

| | | | |
|---|---|---|---|
| Name of Principal Author (Candidate) | Marltan O. Wilson | | |
| Contribution to the Paper | Designed and trained machine learning algorithms, carried out simulations,<br>analyze and interpret the results, and compose manuscript. | | |
| Overall percentage (%) | | | |
| Certification: | This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper. | | |
| Signature | | Date | 15/12/2022 |

## Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

| | | | |
|---|---|---|---|
| Name of Co-Author | David Huang | | |
| Contribution to the Paper | Project conceptualization, research supervision, data analysis, writing (review and editing) | | |
| Signature | | Date | 09/01/2023 |

| | | | |
|---|---|---|---|
| Name of Co-Author | | | |
| Contribution to the Paper | | | |
| Signature | | Date | |

Please cut and paste additional co-author panels here as required.

# Automated anisotropic coarse-graining of polymers using variational autoencoders

Marltan O. Wilson[1] and David M. Huang[1]
*Department of Chemistry, School of Physics, Chemistry and Earth Sciences,*
*The University of Adelaide, Adelaide, South Australia 5005, Australia*

(*Electronic mail: david.huang@adelaide.edu.au)

(Dated: 10 January 2023)

We demonstrate the automated coarse-graining of anisotropic molecules and polymers using an autoencoder neural network. The encoder network in an autoencoder is used to automatically generate a latent space that represents the position and orientation of ellipsoidal coarse-grained sites. The decoder network reconstructs an atomistic configuration from the position and orientations encoded in the latent space. This reconstruction from the latent space has a higher fidelity when compared to reconstruction from the center-of-mass alone. This method of automatic anisotropic coarse-graining creates a straightforward strategy to construct an anisotropic coarse-grained representation of semiconducting polymers with anisotropic subunits, and also provides a back-mapping technique that preserves the probability distribution of the conformation space of the original molecule. The automated anisotropic coarse-graining technique is validated through the ability to construct a coarse-grained representation of a solution-phase hexamer of P(NDI2OD-T2), also known as N2200, that can reproduce the physical observable of its atomistic counterpart. The technique is further validated on the comparatively smaller sexithiophene molecule in the liquid phase. We further show that the optimal number of coarse-grained sites can be determined from the loss versus cost for a given number of coarse-grained sites.

## I. INTRODUCTION:

The recent demand for alternative photovoltaic cells, wearable electronics, and optoelectronic devices have led to intensified research in the area of organic semiconductors.[1–3] This has led to the discovery and utilization of increasingly complex and diverse macromolecules and polymers. It has also become increasingly evident that computational methods, such as molecular dynamics and more recently machine learning, are playing an increasingly large role in material design and discovery.[4–7] However, there are still some limitations on the size and length scale of classical atomistic simulations of materials. Coarse-graining has long been used as a technique to overcome these limitations[8] but to fully utilize a coarse-grained model there needs to be sufficiently accurate, quantifiable, and straightforward back-mapping techniques.

Back-mapping algorithms are important[9] in the field of organic semiconductors because they provide an avenue to study long time-scale properties such as solution-phase aggregation of polymers by running simulations at low resolutions with the possibility of upsampling the system at a later time to study properties such as charge transport or the effects of different functional groups or anisotropy on short-ranged interactions. Many recent breakthroughs in the area of coarse-graining and back mapping came from the integration of machine learning into the field of molecular simulations. The fast-paced growth and development of machine learning tools have increased their popularity in many scientific fields.[10] Autoencoders in particular are popular neural networks developed for data compression problems, in the image-processing sphere,[11] and this machine learning tool has been adapted for uses in the coarse-graining[12] of organic molecules to improve simulation speed and scale. There has been a consistent effort in the attempt to determine the optimal number of coarse-grained sites for generic molecules.[13–16] Unlike traditional methods of coarse-graining, autoencoders do not require a thorough prior understanding of the simulation system, since it is an unsupervised form of machine learning.[17] In general, autoencoders consist of two feedforward neural networks trained together to minimize the data loss between the real data and the data reconstructed from the compressed state. The encoder network is responsible for data compression and in the case of coarse-graining, the encoder network produces the coarse-grained representation of the molecule from the trajectories of the atomistic model obtained from molecular dynamics simulations. On the other hand, the decoder network reconstructs the atomistic trajectory from the coarse-grained representation. This method of coarse-graining attempts to address two major issues in organic semiconductor research. The first is the creation of a coarse-graining methodology that can be compared and optimized without the need for further molecular dynamics simulations. The second problem addressed by the method is its ability to produce a backward map from the coarse-grained representation to the atomistic model.

Even though there have been previous autoencoder models designed to coarse-grain and back-map small molecules to and from isotropic coarse-grain sites,[12] there is still a gap in the knowledge required for coarse-graining macromolecules and polymers into more general ellipsoidal coarse-grain sites accounting for the anisotropy in the mass distribution of different monomers and side-chains.

## II. THEORY

For this work, it is assumed that the computational efficiency of a coarse-grain model decreases linearly with the number of sites, and an optimal coarse-grained representation of a molecule is a model which balances computational efficiency with reconstruction fidelity. Since the neural network loss versus the number of coarse-grained sites is defined on the set of integers, it is defined as continuous at point $b$ if $g(b) = f(b)$, where $g(x)$ is a decay curve fit to the data points on the real interval $(a, b)$ and $f(x)$ is a decay curve fit to the data points on the real interval $[b, c)$, the fit is discontinuous otherwise. A given number of coarse-grained sites $b$ is considered optimal on the interval $(a, c)$ if there is a discontinuity at point $b$ as shown in Fig. 1.

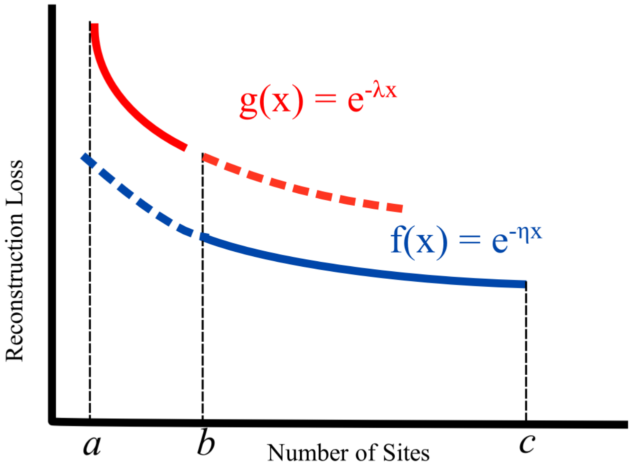


FIG. 1. Diagram showing a typical case where point $b$ is considered an optimal number of coarse-grained sites since $g(b) \neq f(b)$ when $g(x)$ is fitted to the data on the interval $(a, b)$, $f(x)$ is fitted to the data on the interval $[b, c)$, and $g(x)$ and $f(x)$ are both real-valued functions.

A set of mapping functions $\boldsymbol{M}$ are defined such that each fine-grained coordinate $\boldsymbol{r}^n$ is linearly mapped to a unique coarse-grained site $I$ with position $\boldsymbol{R}_I$ and orientation $\boldsymbol{\Omega}_I$ using

$$\boldsymbol{M}_{RI}(\boldsymbol{r}^n) = \boldsymbol{R}_I \tag{1}$$

and

$$\boldsymbol{M}_{\Omega I}(\boldsymbol{r}^n) = \boldsymbol{\Omega}_I, \tag{2}$$

where $\boldsymbol{M}_{RI}$ maps $\boldsymbol{r}^n$ to the centers-of-mass

$$\boldsymbol{R}_I = \frac{\sum_{i\in\zeta_I} m_i \boldsymbol{r}_i}{\sum_{i\in\zeta_I} m_i}, \tag{3}$$

and $\boldsymbol{M}_{\Omega I}$ maps $\boldsymbol{r}^n$ to the principal inertia axes defined by the inertia tensor,

$$\mathbb{I}_{\mathrm{FG},I} = \sum_{i\in\zeta_I} m_i(||\Delta\boldsymbol{r}_i||^2 E - \Delta\boldsymbol{r}_i \Delta\boldsymbol{r}_i^{\mathrm{T}}), \tag{4}$$

where $\Delta\boldsymbol{r}_i = \boldsymbol{r}_i - \boldsymbol{R}_I$ is the position of fine-grained particle $i$ relative to the center-of-mass (coarse-grained particle position), $E$ is the $3 \times 3$ identity matrix and the sums are over the set $\zeta_I$ of fine-grained particles that are mapped onto coarse-grained site $I$. For consistency between the coarse-grained and fine-grained models, the configurational distribution of the coarse-grained model must match that of the fine-grained system on which it is based.

## A. Data preprocessing

The automatic coarse-graining of polymers can take two possible forms:

1. Unconstrained
2. One or more coarse-grained sites per monomer

To determine which method is best suited for a particular polymer, the cost to simulate vs the compression loss must be optimized. In the case of unconstrained coarse-graining, the total number of CG sites is chosen to be less than the number of monomers. The entire polymer is treated as a single macromolecule; that is, for each simulation snapshot, the molecular configuration is flattened into a vector, the center-of-mass is shifted to zero, and the configuration is rotated such that the principal axes of the polymer align with the laboratory frame. The neural network is then unconstrained in allocating atoms to each of the coarse-grained sites. This approach is especially useful for short polymers with simple repeating units. The unconstrained approach can also be used to coarse-grain rigid polymers in which the persistence length is multiple monomers or other cases where it is appropriate to map multiple monomers to a single site. On the other hand, to obtain one or more coarse-grained sites per monomer, the molecular configurations are reshaped to a $P \times S$ matrix, where $P$ is the number of monomer units and $S$ is the number of atoms per monomer, then a similar procedure is followed to center and rotate the polymers in each snapshot with respect to the center-of-mass of each monomer. The neural network is then used to assign a predetermined number of coarse-grained sites to each of the monomer units. For polymers with relatively large repeating units and complex side-chains, it is advantageous to represent the polymer as a $P \times S$ matrix since it increases the number of data points used to train the neural network weights, effectively eliminating the degree of polymerization as a possible source of error.

The neural network method also allows for the integration of prior knowledge into the definition of the coarse-grained sites. A condition can be enforced such that all or some of the coarse-grained sites have the same standard deviation by using the average standard deviation of the specified number of equivalent sites. This condition allows the user to fix the number of CG site types that can be generated independently of the overall number of coarse-grained sites specified. The difference in the reconstruction fidelity as a function of CG site types can also be used to determine the optimal anisotropic coarse-grained representation of any molecule.

## B. Encoder algorithm

The encoder network is constructed such that

1. The mass of the coarse-grain site is taken as the sum of the masses of the contributing atoms from the fine-grain model.

2. The inertia tensor of each ellipsoidal site is derived from the average fluctuations of the contributing atoms about the center-of-mass of the coarse-grain site, which will be further explained in the following sections.

3. No atom from the fine-grained model is mapped to more than one coarse-grained site.

The first and second conditions outlined above are satisfied by using Eqns. (3) and (4) as target values for the construction of a normal distribution with mean $\mu_I$ and standard deviation $\sigma_I$. The mean of the probability distribution of the mass-weighted positions of the atoms corresponds to the mean of the center-of-mass defined in Eqn. (3) and the standard deviation of the 3D joint probability distribution of the mass-weighted atom positions generates the principal axes of the coarse-grained site defined by Eqn. (4). For the case where more than one center-of-mass is defined corresponding to multiple coarse-grained sites per molecule, the probability distribution becomes a multi-modal distribution. However, straightforward enforcement of the third condition requires no mixing between the modes of the distribution, which would require assigning an atom to the coarse-grained site of the highest probability, according to

$$\boldsymbol{Z_i} = \text{one\_hot}(\underset{I}{\text{argmax}}\{\log \pi_{iI}\}) \tag{5}$$

where $\boldsymbol{Z_i}$ is a categorical variable and $\pi_{iI}$ is the probability that atom $i$ is assigned to coarse-grain site $I$. However, this argmax function would make the neural network nondifferentiable and prevent learning through backward propagation.[18] To enforce the first condition without trying to backpropagate through a non-differentiable layer, the Gumbel-softmax reparametrization[18] trick is used to approximate an argmax function. Gumbel-softmax reparameterization allows a variational autoencoder to approximate sampling from a discrete latent space through the introduction of a neural network temperature variable giving the $I$th element of $\boldsymbol{Z_i}$ as

$$Z_{iI} = \frac{\exp((G_{iI} + \log \pi_{iI})/\tau)}{\sum_j^n \exp((G_{jI} + \log \pi_{jI})/\tau)} \tag{6}$$

Here, $G_{iI}$ is a sampled from the standard Gumbel distribution and $\tau$ is the temperature variable, such that as $\tau \to 0$ the softmax calculations smoothly approach argmax and $\boldsymbol{Z_i}$ approximates a one-hot vector. By initializing the neural network with a sufficiently large temperature variable, each atom in a molecule can transition across all available coarse-grained sites.[12] The subsequent annealing process lowers the temperature gradually ensuring that each atom is mapped to the optimal coarse-grained site in such a way that the overall coarse-grained model reproduces the mass distribution of the all-atom model. The encoder network performs linear transformations assigning the atomistic configurations to the centers-of-mass and the inertia tensor of the coarse-grained ellipsoid. By retaining the mass distribution along each of the principal axes, the model generalizes spherically symmetric coarse–grain sites to anisotropic ellipsoidal sites. As the neural network temperature variable decreases, each atom only contributes to the calculation of the mean of a single coarse-grained site and the fluctuation of the atom about the mean position defines the standard deviation and by extension the principal axes of the coarse-grained site. Since each molecular trajectory is fixed to the molecular center-of-mass, atoms close to the molecular center-of-mass will have smaller fluctuations and will be the first to anneal into their final position. Atoms at the far ends of a polymer or side-chains will fluctuate more widely and will require more data to produce consistent results for their coarse-grained representation. The latent space of the encoder network provides a set of positions $\boldsymbol{R}_I$ and orientations $\boldsymbol{\Omega}_I$ for each coarse-grained site.

### C. Decoder algorithm

The decoder is responsible for the reconstruction of the atomistic trajectories from the coarse-grained latent space representation.[19] In the automatic anisotropic coarse-graining method, the reconstruction is done using two pieces of information, the center-of-mass of each coarse-grained site as well as the inertia tensor which describes an ellipsoidal mass distribution about each of the coarse-grained center-of-mass. Compared to a spherical coarse-grained model, reconstruction fidelity is improved for the anisotropic model since it uses information about the inertia tensor in the decoding process. This additional reconstruction fidelity is important for organic semiconductors since back mapping is an important tool to understand charge transfer in polymer aggregates.[20]

The loss function of the autoencoder has two sources contributing to the total loss, The first being the reconstruction loss and the second being the reparameterization loss. The reconstruction loss can be further broken down into the reconstruction of one–, two– and three–body contributions, that is, the reconstruction of the atom positions, bonds, and angles respectively. This is achieved through the use of a regression loss function namely, the mean squared error,

$$L_{\text{recon}} = ||\Gamma_{\text{D}}(\Gamma_{\text{E}}(\boldsymbol{X}, \tau, \boldsymbol{G})) - \boldsymbol{X}||^2 \tag{7}$$

where $\Gamma_{\text{D}}$ and $\Gamma_{\text{E}}$ are the decoder and encoder network function, and $\tau$ and $\boldsymbol{G}$ are the neural network temperature variable and the sampled Gumbel distribution, respectively. On the other hand, since the reconstruction of the trajectories is probabilistic, the reparametrization error minimizes the distance between the true distribution of the atomic positions and the sampled distribution used for the reconstruction. This reparameterization error is constructed as the evidence lower bound.[21] The variational autoencoder aims to maximize the likelihood of recovering the data from the latent representation, $p(\boldsymbol{Z}|\boldsymbol{X})$, where $\boldsymbol{Z}$ is the latent representation and $\boldsymbol{X}$ is the data. given the input data has true distribution $p(\boldsymbol{X})$ and the latent representation has distribution $q(\boldsymbol{Z})$, the evidence lower bound is defined as

$$\mathrm{ELBO} = \mathbb{E}_q \left[ \log \frac{p(\boldsymbol{X}|\boldsymbol{Z})}{q(\boldsymbol{Z})} \right], \tag{8}$$

where $\mathbb{E}_q$ is the expectation. The total loss is calculated using

$$L_{\mathrm{total}} = L_{\mathrm{recon}} - 0.5 \times \mathrm{ELBO} \tag{9}$$

The gradient descent algorithm was implemented using the Adam optimizer.[22] A schematic of the autoencoder architecture is shown in Fif. 2.

### D. Atomistic simulation and coarse-grained potential

$\alpha$-Sexithiophene and a hexamer of the polymer P(NDI2OD-T2), also known as N2200, were chosen as test molecules to demonstrate the different capabilities of the anisotropic autoencoder. Atomistic MD simulations were done using the molecular dynamics software package LAMMPS (version 20NOV19).[23–25] The OPLS-AA force field[26–29] and a cut-off of 10 Å were used for the simulation of 250 sexithiophene molecules in the isothermal-isobaric (NPT) ensemble with the pressure set at 1 atm and temperature of 680 K.[30] A molecular dynamics simulation of a single N2200 hexamer in a solution of 14680 chloroform molecules was carried out at 300 K and 1 atm in the NPT ensemble with OPLS-AA force field and a cutoff of 11 Å. For all atomistic simulations hydrogen bonds were constrained with the SHAKE algorithm,[31] long-ranged electrostatic interactions were calculated with the particle–particle particle–mesh (PPPM) method,[32,33] and the temperature and pressure controlled by a Nosé–Hover thermostat and barostat.[34,35] The N2200 hexamer in chloroform solution was equilibrated for 1 ns and then simulations were carried out with the time step set to 2 fs and simulations ran for 1 ns. The sexithiophene simulations were 25 ns long with a timestep of 1 fs. The last 20 ns of the simulation data was used for parameterization of the coarse-grained potential and calculation of structural distributions.

To use the coarse-grained models for molecular dynamics simulations, the coarse-grained potential was fitted by using the instantaneous forces and torques to train a neural network potentntial with explicit inclusion of dihedral angles between nearest-neighbor anisotropic monomers. This corresponds to the force-matching condition in the AFM-CG method, which is required for thermodynamic consistency. A schematic of the neural network used in the force matching procedure is shown in Fig. 3.

Each monomer had a ghost atom attached at off-center positions for the definition of bonds between ellipsoids Fig. 4. This ensures that forces and torques are correctly applied to the anisotropic particle and not just the center-of-mass of the monomer. The bond length, bond angle, and dihedral potentials are given by

$$U_{\mathrm{bond}} = K_{\mathrm{B}}(b - b_0)^2 \tag{10}$$

$$U_{\mathrm{angle}} = K_{\mathrm{A}}(\theta - \theta_0)^2 \tag{11}$$

$$\begin{aligned} U_{\mathrm{dihedral}} &= \frac{1}{2}K_1[1 + cos(\phi)] + \frac{1}{2}K_{\mathrm{D}}[1 - cos(2\phi)] \\ &+ \frac{1}{2}K_3[1 + cos(3\phi)] + \frac{1}{2}K_4[1 - cos(4\phi)] \end{aligned} \tag{12}$$

where $b$ and $b_0$ are the instantantous and equilibrium bond lengths, respectively, $\theta$ and $\theta_0$ are the instantaneous and equilibrium bond angles, respectively, $\phi$ is the dihedral angle, $K_{\mathrm{B}}$ and $K_{\mathrm{A}}$ are the bond and three-body angle potential parameter, respectively, and $K_1$, $K_{\mathrm{D}}$, $K_3$, and $K_4$ are the coefficient of the OPLS cosine expansion of the dihedral potential. Nonbonded interactions were defined between particles separated by one bond (1–2 interactions).

Fitting the forces to the derivative of the potential was done using TensorFlow's gradient descent algorithm and the derivative of the potential was implemented using TensorFlow's GradientTape function to evaluate the computational derivative.[36,37] The hyperparameters for the neural network was fitted using modified a modified version of the Behler symmetry functions.[38] When fitting the coarse-grained potential using the neural network, each coarse-grain site is mapped to an invariant vector representation $\boldsymbol{D}_{IJ}$ which is defined in terms of the position and orientations of particles $I$ and $J$ and is given by

$$\begin{aligned} \boldsymbol{D}_{IJ} = \{ & R_{IJ}, \boldsymbol{R}_{IJ} \cdot \boldsymbol{\Omega}_{I,1}, \boldsymbol{R}_{IJ} \cdot \boldsymbol{\Omega}_{I,2}, \boldsymbol{R}_{IJ} \cdot \boldsymbol{\Omega}_{I,3}, \\ & \boldsymbol{R}_{IJ} \cdot \boldsymbol{\Omega}_{J,1}, \boldsymbol{R}_{IJ} \cdot \boldsymbol{\Omega}_{J,2}, \boldsymbol{R}_{IJ} \cdot \boldsymbol{\Omega}_{J,3}, \\ & \boldsymbol{\Omega}_{I,1} \cdot \boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,1} \cdot \boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,1} \cdot \boldsymbol{\Omega}_{J,3}, \\ & \boldsymbol{\Omega}_{I,2} \cdot \boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,2} \cdot \boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,2} \cdot \boldsymbol{\Omega}_{J,3}, \\ & \boldsymbol{\Omega}_{I,3} \cdot \boldsymbol{\Omega}_{J,1}, \boldsymbol{\Omega}_{I,3} \cdot \boldsymbol{\Omega}_{J,2}, \boldsymbol{\Omega}_{I,3} \cdot \boldsymbol{\Omega}_{J,3} \}, \end{aligned} \tag{13}$$

where $\boldsymbol{R}_I$, $\boldsymbol{R}_J$, $\boldsymbol{\Omega}_I$, and $\boldsymbol{\Omega}_J$ are obtained from the encoder latent space and $\boldsymbol{R}_{IJ} \equiv \boldsymbol{R}_I - \boldsymbol{R}_J$. The neighbourhood of particle $I$ can then be represented by a unique fingerprint $\mathbb{D}_I$ which is obtained from the concatenation of all the $\boldsymbol{D}_{IJ}$ vectors in the neighbourhood of particle $I$. The prior repulsive potential can then be represented by the simply as

$$U_{\mathrm{prior},I} = \sum_{J \neq I} B_1 \sigma_{\mathrm{c}} (\mathbb{D}_I)^{-B_2}, \tag{14}$$

where $\sigma_{\mathrm{c}}$ is a neural-network function and $B_1$ and $B_2$ are trainable parameters. The total potential $U$ can then be written as a sum over all $U_I$ contribution given as

$$U_I = U_{\mathrm{NN},I} + U_{\mathrm{prior},I} + U_{\mathrm{bond},I} + U_{\mathrm{angle},I} + U_{\mathrm{dihedral},I}. \tag{15}$$

and

$$U = \sum_{I=1}^{N} U_I \tag{16}$$

A more indepth discussion of the force matching neural network architecture can be found in the supporting information.

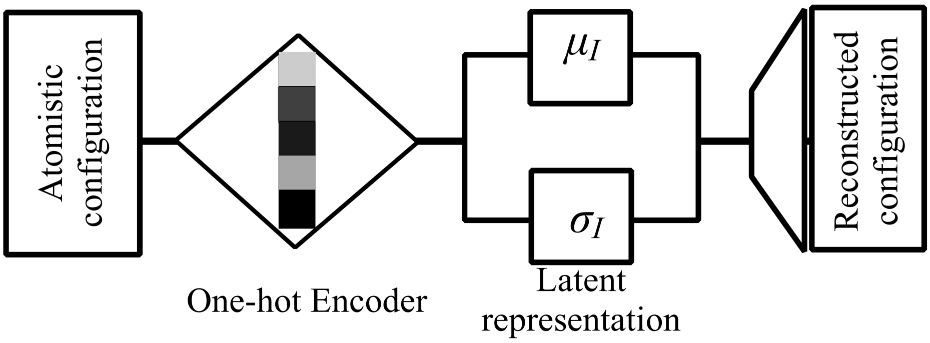


FIG. 2. Schematic of the neural network architecture used to map polymer atomistic configurations to a discrete latent space parameterized by the mean and standard deviation of a multimodal joint ellipsoidal distribution.

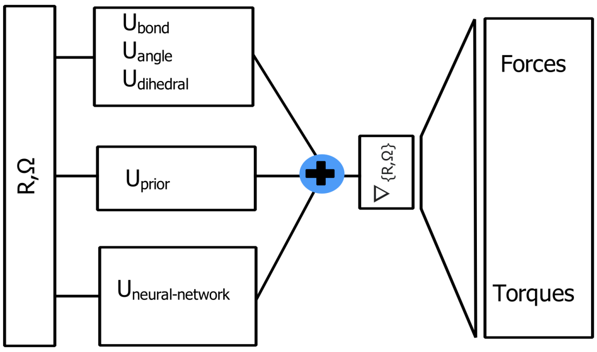


FIG. 3. Schematic of the neural network used to fit the coarse-grain potential.

The interaction between the N2200 ellipsoids and the spherical solvent particles as well as the solvent–solvent interactions were derived from the same procedure above.

A six-site coarse-grained representation was used for the coarse-grained simulation of both N2200 and sexithiophene. The simulations were done in the canonical ensemble (NVT) to match the density of the atomistic simulations. The sexithiophene coarse-grained simulations were performed at 590 and 680 K with 250 molecules. A single-site model of sexithiophene was also parameterized under the same conditions. The CG simulations with the N2200 hexamer and 14680 isotropic chloroform solvents were done at 300 K.[39] The N2200 hexamer in chloroform solution as well as the sexitiophene coarse-grained simulations were 25 ns long with the last 20 ns used for the calculation of structural distributions.

## III. RESULTS AND DISCUSSION

### A. $\alpha$-Sexithiophene

Sexithiophene shown in Fig. 5 has been researched as a promising material for organic photovoltaics[40] and organic light-emitting diodes.[41,42] There has been significant research into controlling the orientation of sexithiophene deposited on substrates.[43,44] Sexithiophene was used to demonstrate the unconstrained coarse-graining ability of the anisotropic autoencoder. Sexithiophene coarse-grained to a single ellipsoid does not capture the backbone flexibility or any of the thiophene-thiophene torsional configurations. The neural network loss was calculated for different numbers of coarse-grained sites ranging from one to six. The plot of loss versus the number of sites in Fig. 6 shows a notable decrease in reconstruction loss when more than one coarse-grained site is used to model sexithiophene, whereas there is a smaller decrease in reconstruction loss when the number of sites increases from two to six. Since sexithiophene consists of six monomers, a steep decrease in the neural network reconstruction loss between five and six coarse-grain sites is expected. With six available coarse-grain sites the neural network can more accurately reconstruct the mass distribution changes due to the rotation of the monomers about the thiophene–thiophene bonds as shown in Fig. 7.

The six-site neural network model of sexithiophene captures the structural variations in the liquid and liquid crystal phases as shown in Fig. 8. The six-site coarse-grain model of sexithiophene outperforms the single-site model when comparing the orientational order parameter in the liquid crystal phase (Fig. 9). However, there are only small differences between the six-site and the single-site model when comparing the center of mass radial distribution function (Fig. 10).

The one-site sexithiophene model had a 132 × speed-up compared to the all-atom model while the six-site sexithiophene model had a 17 × speed-up compared to the all-atom model.

### B. P(NDI2OD-T2)

Poly[N,N′-bis(2-octyldodecyl)naphthalene-1,4,5,8-bis(dicarboximide)-2,6-diyl]-alt-5,5′-(2,2′-bithiophene) (P(NDI2OD-T2)), also known as N2200, is a copolymer of naphthalene diimide (NDI) and bithiophene units with alkyl side chains. There have been significant interest in N2200 as an organic semiconductor.[45–49] It is considered

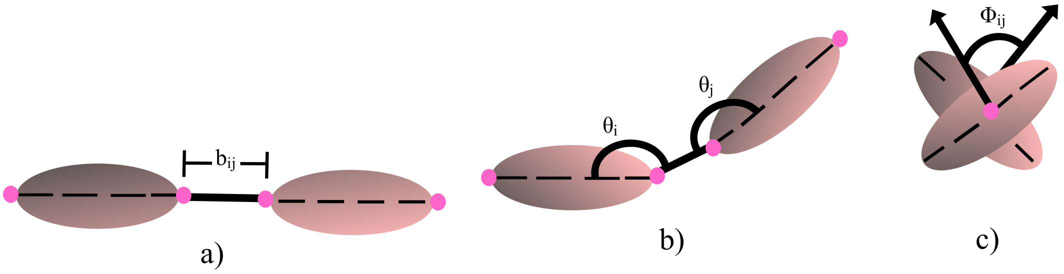


FIG. 4. Schematic of the (a) bonds, (b) angles, and (c) dihedrals as defined for the anisotropic polymer models.

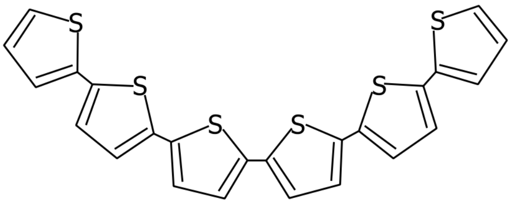


FIG. 5. Chemical structure of sexithiophene

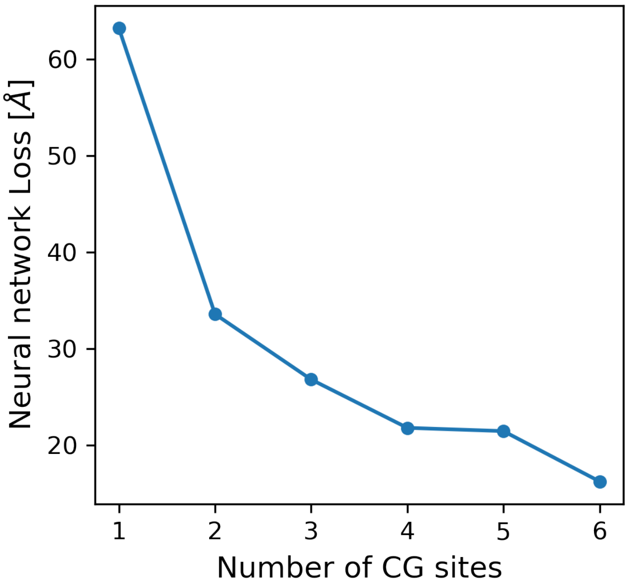


FIG. 6. Neural network reconstruction loss versus the number of coarse-grained sites when sexithiophene is mapped to between one and six coarse-grain sites.

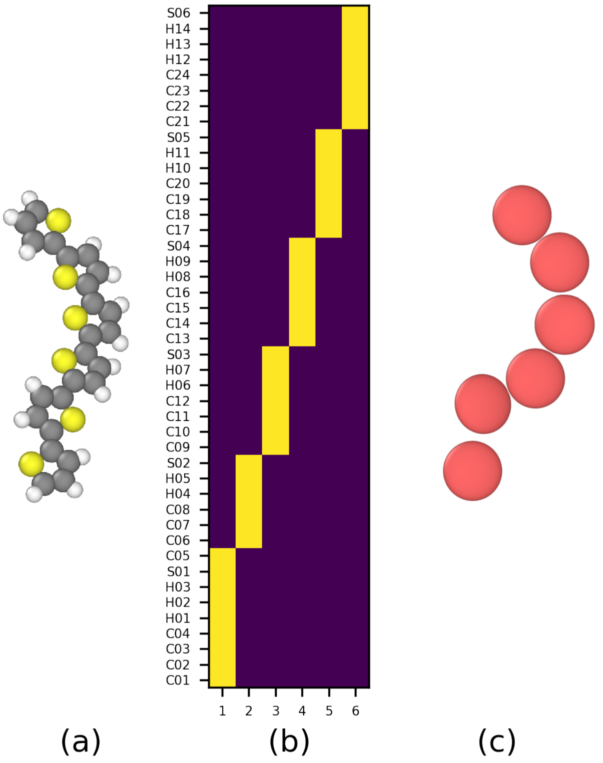


FIG. 7. Six-site coarse-grained model of sexithiophene. The (a) atomistic configuration was mapped to the latent space using the (b) learned encoding, producing a mapping to the (c) position and orientation of the coarse-grained sites. The rows of the encoding matrix in (b) represent each atom and the columns are the available coarse-grained sites.

one of the best organic polymer acceptors due to its high electron mobility[50] and narrow band gap.[51] N2200 has had recent success in organic solar cell applications[52] and energy storage in the form of capacitors.[46] N2200 was chosen to demonstrate how well the anisotropic autoencoder handles one or more coarse-grained sites per monomer. This also provides an opportunity to see how well the neural network method handles flexible alkyl side chains and an aromatic backbone. Plots of the neural network loss versus the number of coarse-grained sites are shown in Fig. 11. These plots showed several discontinuities where the loss between consecutive numbers of coarse-grained sites showed a larger decrease than for the pair before or the pair after.

The unconstrained allocation of coarse-grained sites for the N2200 hexamer starts with a relatively high error which can

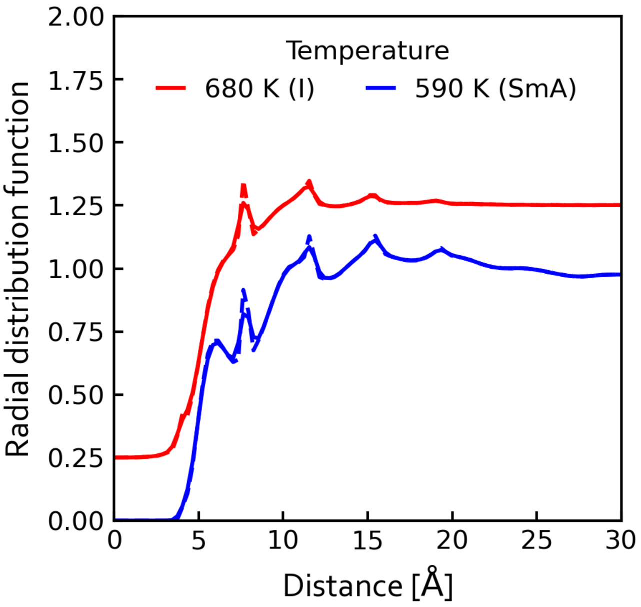


FIG. 8. All-atom (solid lines) and six-site (dashed lines) coarse-grained monomer-monomer radial distribution function for sexithiophene in the isotropic phase (680 K) and the Smectic-A phase (590 K).

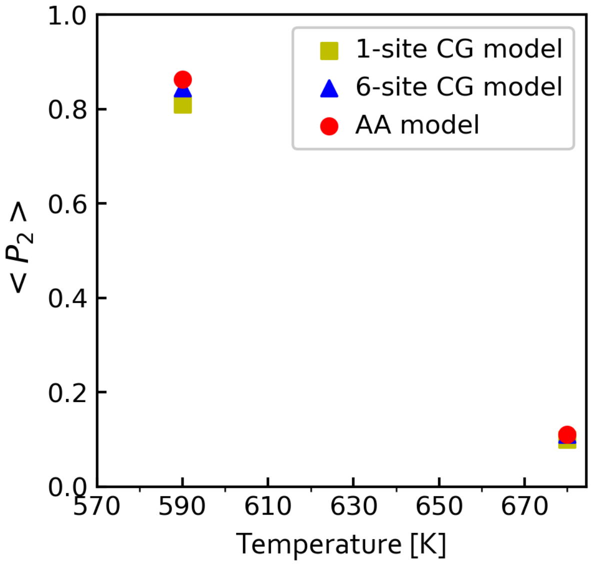


FIG. 9. Orientational order parameter for all-atom, six-site coarse-grain and one-site coarse-grain models of sexithiophene at 590 and 680 K

be attributed to the attempt to represent a flexible molecule as a rigid ellipsoid. Even though there is a significant drop in the reconstruction error between one and five sites, the trend still follows the expected exponential decay that would be expected just from adding more complexity to the model. The only significant feature that is observed on the interval $[1, 12]$ is a discontinuity in the decay trend between five and

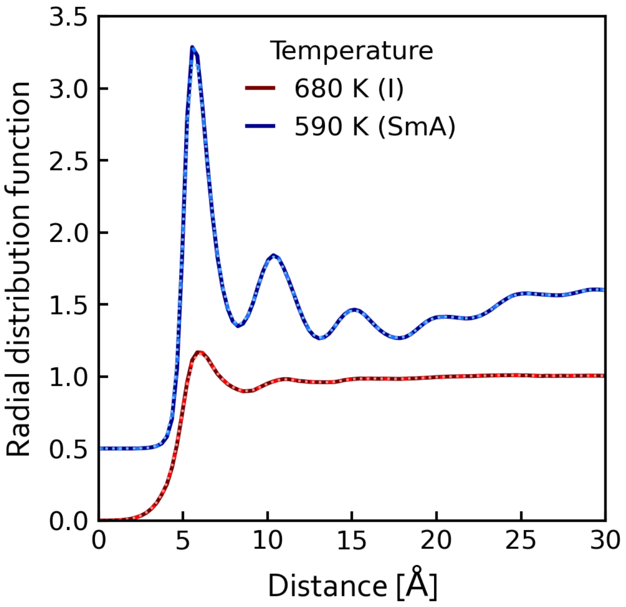


FIG. 10. All-atom (solid lines), six-site coarse-grained (dashed lines), and single-site coarse-grained (dotted lines) center-of-mass radial distribution function for sexithiophene in the isotropic phase (680 K) and the smectic-A phase (590 K).

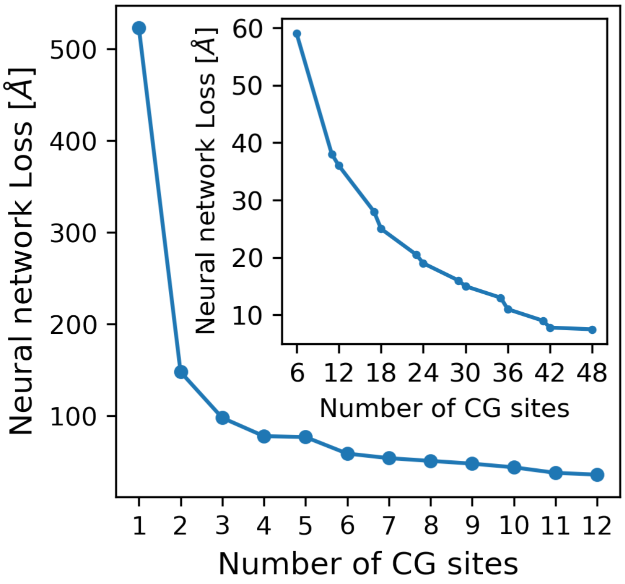


FIG. 11. Neural network reconstruction loss versus number of coarse-grained sites for the case where the number of sites is less than twice the number of monomers for the N2200 hexamer (main plot) and for one or more sites per monomer for the N2200 hexamer. Lines connecting the data points are solely for visualizing the trend between adjacent data points.

six coarse-grained sites. As expected there is a significant drop in the reconstruction error when each of the monomers in the polymer is assigned to individual sites. There is a discontinuity in the plot of reconstruction loss versus the number of sites when eighteen and forty-two coarse-grained ellipsoids are allocated. The coarse-grained model with three sites per monomer separated the backbone of the polymer from the sidechains. The allocation of the atoms associated with forty-two sites or seven sites per monomer is shown in Fig. 12. The anisotropic autoencoder was able to group each branch of the alkyl side chains into an ellipsoid while also grouping the naphthalene diimide (NDI) and bithiophene units into individual ellipsoids. Fig. 11(inset) shows a trend of increased reduction in the loss for every six additional sites added to the polymer. By observing how the neural network allocates the atoms for the seven-site model, a priori information can be added to the neural network by enforcing a set of only three unique ellipsoid types for the seven available sites: that is, four ellipsoids assigned to the first type, two to the second and one to the third. The results are shown in the color-coding of Fig. 12. This added flexibility can significantly simplify the output of the neural network latent space with less than 2 % increase in the reconstruction error.

The comparison of the center-of-mass radius of gyration for the all-atom, six-site coarse-grained, and back-mapped models is shown in Fig. 13. There is a close match between the all-atom and the back-mapped models, the discrepancy between the coarse-grained and all-atom models can be attributed to the method of calculation, where it was assumed that the entire mass of each monomer acts at the center-of-mass of the coarse-grain ellipsoid instead of distributed over the entire volume.

The six-site representation of N2200 hexamer provides an opportunity for high fidelity backmapping but lacks the flexibility to fully capture the radius of gyration or the solute-solvent distributions as shown in Fig. 14. A higher resolution model with allocation for side-chain interactions would be required to study polymer–solvent interactions. The six-site N2200 model had a 161 × speed-up compared to the all-atom model

## CONCLUSIONS

We have shown that an unsupervised machine-learning approach can be used to coarse-grain large molecules and polymers using either an unconstrained approach or by prescribing one or more sites per monomer. With the inclusion of anisotropic mass distribution data for the coarse-grained sites, the autoencoder was able to increase the reconstruction fidelity of large molecules with anisotropic mass distribution. The anisotropic feature is especially highlighted with the organic semiconducting polymer sexithiophene and N2200 since they both contain anisotropic monomer units. Additionally, the automatic anisotropic coarse-graining method provides the ability to specify the number of unique types of ellipsoids independently of the specified number of coarse-grained sites. This feature simplifies the coarse-grained representation of polymers with complex monomers such as N2200.

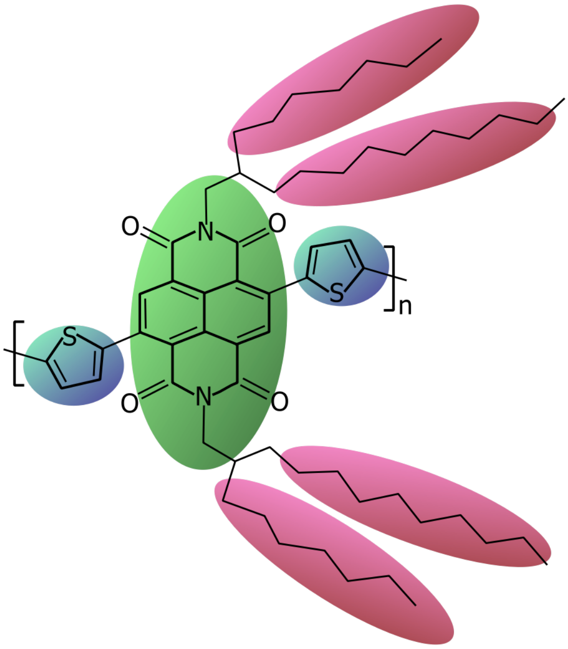


FIG. 12. Neural network representation of coarse-grained N2200 where the number of coarse-grained sites is set to seven disjoint sets and the color- coding represents sites with the same inertia tensor.

## ACKNOWLEDGMENTS

This work was supported by the Australian Research Council under the Discovery Projects funding scheme (DP190102100). This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government, and from the University of Adelaide's Phoenix High-Performance Computing service.

## REFERENCES

[1] P. Meredith, W. Li, and A. Armin, "Nonfullerene acceptors: A renaissance in organic photovoltaics?" Adv. Energy Mater. **10**, 2001788 (2020).

[2] X. Yang, L. Ding, *et al.*, "Organic semiconductors: commercialization and market," J. Semicond **42**, 090201 (2021).

[3] K. Yu, S. Rich, S. Lee, K. Fukuda, T. Yokota, and T. Someya, "Organic photovoltaics: Toward self-powered wearable electronics," Proc. IEEE **107**, 2137–2154 (2019).

[4] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, and M. Lei, "Machine learning in materials science," InfoMat **1**, 338–358 (2019).

[5] C. H. Chan, M. Sun, and B. Huang, "Application of machine learning for advanced material prediction and design," EcoMat , e12194 (2022).

[6] S. Xiao, R. Hu, Z. Li, S. Attarian, K.-M. Björk, and A. Lendasse, "A machine-learning-enhanced hierarchical multiscale method for bridging from molecular dynamics to continua," Neural Comput. App. **32**, 14359–14373 (2020).

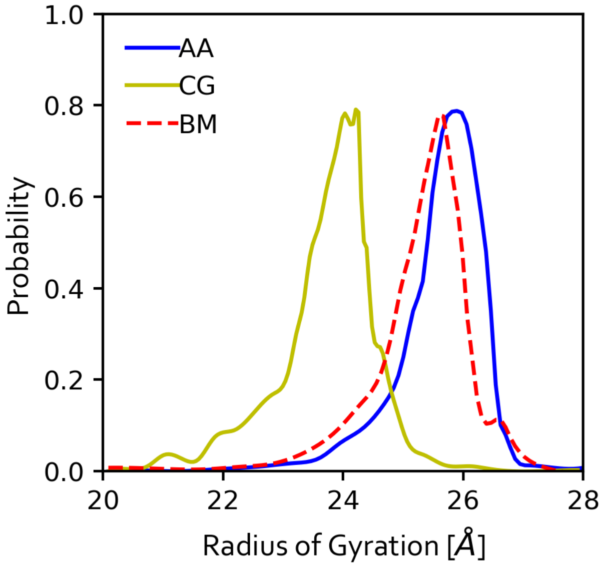


FIG. 13. A comparison of (a) the distribution of the center-of-mass radius of gyration of the all-atom (AA), six-site coarse-grained (CG), and the back mapped (BM) model of the N2200 hexamer in chloroform solution at 300 K.

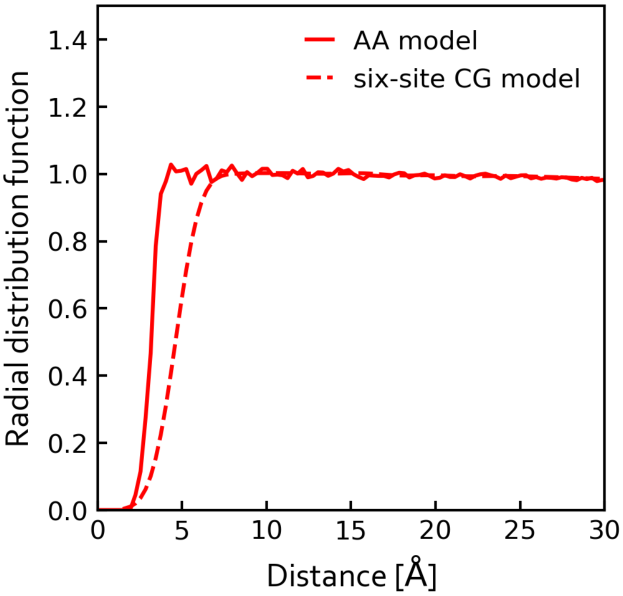


FIG. 14. The monomer-solvent center-of-mass radial distribution function for the all-atom and six-site coarse models of the N2200 hexamer in chloroform solution at 300 K.

[7] T. Okamoto, S. Kumagai, E. Fukuzaki, H. Ishii, G. Watanabe, N. Niitsu, T. Annaka, M. Yamagishi, Y. Tani, H. Sugiura, *et al.*, "Robust, high-performance n-type organic semiconductors," Sci. Adv. **6**, eaaz0632 (2020).

[8] J. Jin, A. J. Pak, A. E. Durumeric, T. D. Loose, and G. A. Voth, "Bottom-up coarse-graining: Principles and perspectives," J. Chem. Theory Comput. **18**, 5759–5791 (2022).

[9] W. Li, C. Burkhart, P. Polińska, V. Harmandaris, and M. Doxastakis, "Backmapping coarse-grained macromolecules: An efficient and versatile machine learning approach," J. Chem. Phys. **153**, 041101 (2020).

[10] K. K. Bejagam, S. Singh, Y. An, and S. A. Deshmukh, "Machine-learned coarse-grained models," J. Phys. Chem. Lett. **9**, 4667–4672 (2018).

[11] Y. Zhang, "A better autoencoder for image: Convolutional autoencoder," in *ICONIP17-DCEC. Available online: http://users. cecs. anu. edu. au/Tom. Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58. pdf (accessed on 23 March 2017)* (2018).

[12] W. Wang and R. Gómez-Bombarelli, "Coarse-graining auto-encoders for molecular dynamics," npj Comput. Mater. **5**, 1–9 (2019).

[13] A. V. Sinitskiy, M. G. Saunders, and G. A. Voth, "Optimal number of coarse-grained sites in different components of large biomolecular complexes," J. Phys. Chem. B **116**, 8363–8374 (2012).

[14] M. Li, J. Z. Zhang, and F. Xia, "Constructing optimal coarse-grained sites of huge biomolecules by fluctuation maximization," J. Chem. Theory Comput. **12**, 2091–2100 (2016).

[15] Z. Wu, Y. Zhang, J. Z. Zhang, K. Xia, and F. Xia, "Determining optimal coarse-grained representation for biomolecules using internal cluster validation indexes," J. Comput. Chem. **41**, 14–20 (2020).

[16] A. E. P. Durumeric and G. A. Voth, "Adversarial-residual-coarse-graining: Applying machine learning theory to systematic molecular coarse-graining," J. Chem. Phys. **151**, 124110 (2019).

[17] C. Doersch, "Tutorial on variational autoencoders," arXiv (2016).

[18] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," arXiv (2016).

[19] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "Tensorflow distributions," arXiv (2017).

[20] N. Rolland, M. Modarresi, J. F. Franco-Gonzalez, and I. Zozoulenko, "Large scale mobility calculations in pedot (poly (3, 4-ethylenedioxythiophene)): Backmapping the coarse-grained martini morphology," Comput. Materi. Sci. **179**, 109678 (2020).

[21] A. Asperti and M. Trentin, "Balancing reconstruction error and kullback-leibler divergence in variational autoencoders," IEEE Acc. **8**, 199440–199448 (2020).

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv , 1412.6980 (2014).

[23] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," J. Comput. Phys. **117**, 1–19 (1995).

[24] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington, "Implementing molecular dynamics on hybrid high performance computers – Particle–particle particle–mesh," Comput. Phys. Commun. **183**, 449–459 (2012).

[25] W. M. Brown, P. Wang, S. J. Plimpton, and A. N. Tharrington, "Implementing molecular dynamics on hybrid high performance computers – short range forces," Comput. Phys. Commun. **182**, 898–911 (2011).

[26] W. L. Jorgensen and J. Tirado-Rives, "The opls [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin," J. Am. Chem. Soc. **110**, 1657–1666 (1988).

[27] R. C. Rizzo and W. L. Jorgensen, "OPLS all-atom model for amines: Resolution of the amine hydration problem," J. Am. Chem. Soc. **121**, 4827–4836 (1999).

[28] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, "Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids," J. Am. Chem. Soc. **118**, 11225–11236 (1996).

[29] W. L. Jorgensen and N. A. McDonald, "Development of an all-atom force field for heterocycles. Properties of liquid pyridine and diazenes," J. Mol. Struct. THEOCHEM **424**, 145–155 (1998).

[30] A. Pizzirusso, M. Savini, L. Muccioli, and C. Zannoni, "An atomistic simulation of the liquid-crystalline phases of sexithiophene," J. Mater. Chem. **21**, 125–133 (2011).

[31] J.-P. Ryckaert, G. Ciccotti, and H. J. Berendsen, "Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes," J. Comput. Phys. **23**, 327–341 (1977).

[32] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (CRC Press, 1998).

[33] M. L. Price, D. Ostrovsky, and W. L. Jorgensen, "Gas-phase and liquid-state properties of esters, nitriles, and nitro compounds with the OPLS-AA force field," J. Comput. Chem. **22**, 1340–1352 (2001).

[34] W. G. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," Phys. Rev. A **31**, 1695 (1985).

[35] S. Nosé, "A molecular dynamics method for simulations in the canonical ensemble," Mol. Phys. **52**, 255–268 (1984).

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symp. Oper. Syst. Des. Implement. OSDI 16* (2016) pp. 265–283.

[37] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras (2015).

[38] J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," J. Chem. Phys. **134**, 074106 (2011).

[39] B. J. Boehm, C. R. McNeill, and D. M. Huang, "Competing single-chain folding and multi-chain aggregation pathways control solution-phase aggregate morphology of organic semiconducting polymers," Nano. **14**, 18070–18086 (2022).

[40] J. Sakai, T. Taima, and K. Saito, "Efficient oligothiophene: fullerene bulk heterojunction organic photovoltaic cells," Org. Elect. **9**, 582–590 (2008).

[41] C. Heck, T. Mizokuro, M. Misaki, R. Azumi, and N. Tanigaki, "Oriented polyfluorene films dye-doped for whitening of polarized electroluminescent devices," Jap. J. App. Phys. **50**, 04DK20 (2011).

[42] C. Heck, T. Mizokuro, and N. Tanigaki, "White polarized electroluminescence devices by dye deposition on oriented polyfluorene films," App. Phys. Exp. **5**, 022103 (2012).

[43] T. Matsushima and H. Murata, "Enhanced charge-carrier injection caused by molecular orientation," App. Phys. Let. **98**, 121 (2011).

[44] T. Mizokuro, C. Heck, and N. Tanigaki, "Orientation of $\alpha$-sexithiophene on friction-transferred polythiophene film," J. Phys. Chem.B **116**, 189–193 (2012).

[45] J. Yuan, W. Guo, Y. Xia, M. J. Ford, F. Jin, D. Liu, H. Zhao, O. Inganäs, G. C. Bazan, and W. Ma, "Comparing the device physics, dynamics and morphology of polymer solar cells employing conventional pcbm and non-fullerene polymer acceptor n2200," Nano Energy **35**, 251–262 (2017).

[46] B. A. Wavhal, M. Ghosh, S. Sharma, S. Kurungot, and S. Asha, "A high-voltage non-aqueous hybrid supercapacitor based on the n2200 polymer supported over multiwalled carbon nanotubes," Nano. **13**, 12314–12326 (2021).

[47] G. Wen, X. Zou, R. Hu, J. Peng, Z. Chen, X. He, G. Dong, and W. Zhang, "Ground-and excited-state characteristics in photovoltaic polymer n2200," RSC Adv. **11**, 20191–20199 (2021).

[48] Y. Yan, Y. Liu, J. Zhang, Q. Zhang, and Y. Han, "Optimization of local orientation and vertical phase separation by adding a volatilizable solid additive to the j51: N2200 blend to improve its photovoltaic performance," J. Materi. Chem. C **9**, 3835–3845 (2021).

[49] C. Ren, Y. He, S. Li, Q. Sun, Y. Liu, Y. Wu, Y. Cui, Z. Li, H. Wang, Y. Hao, *et al.*, "Double electron transport layers for efficient and stable organic-inorganic hybrid perovskite solar cells," Org. Elect. **70**, 292–299 (2019).

[50] H. Yan, Z. Chen, Y. Zheng, C. Newman, J. R. Quinn, F. Dötz, M. Kastler, and A. Facchetti, "A high-mobility electron-transporting polymer for printed transistors," Nat. **457**, 679–686 (2009).

[51] C. Mu, P. Liu, W. Ma, K. Jiang, J. Zhao, K. Zhang, Z. Chen, Z. Wei, Y. Yi, J. Wang, *et al.*, "High-efficiency all-polymer solar cells based on a pair of crystalline low-bandgap polymers," Adv. Mater. **26**, 7224–7230 (2014).

[52] K. Wang, S. Dong, K. Zhang, Z. Li, J. Huang, and M. Wang, "Improving the fill factor of n2200-based all polymer solar cells by introducing eppdi as a solid additive," Org. Elect. **99**, 106319 (2021).

# Supplementary Material:
# Automated anisotropic coarse-graining of polymers using variational autoencoders

Marltan O. Wilson and David M. Huang

*Department of Chemistry, School of Physics, Chemistry and Earth Sciences,*
*The University of Adelaide, Adelaide, South Australia 5005, Australia*

## CONTENTS

## SI. FORCE MATCHING ALGORITHM

The set of $G_I^1$ and $G_I^5$ symmetry functions used to construct the local environment of coarse-grained particle $I$ have the functional form

$$G_I^1 = \sum_{J \neq I} g_{\mathrm{c}}(R_{IJ}). \tag{S1}$$

Where $R_{IJ}$ is the separation distance between particle $I$ and $J$ and $g_{\mathrm{c}}$ is a cut-off function with hyperparameter $R_{\mathrm{c}}$ and

$$G_I^5 = \sum_{J \neq I} \prod_{\mu=1}^{M} 2^{1-\nu} \left(1 + \lambda \cos\theta_{IJ,\mu}\right)^{\nu} e^{-\eta (R_{IJ}-R_s)^2} g_{\mathrm{c}}(R_{IJ}). \tag{S2}$$

where $\lambda \in \{-1,1\}$ and $R_{\mathrm{s}}$, $\nu$, and $\eta$ are tunable hyperparameters and $\{\cos\theta_{IJ,\mu}\}$, is the set of machine-learned collective variables with the same properties as the angular component of the underlying potential and $M$ is the total number of machine-learned angular variables. The hyperparameters used for sexithiophene and the N2200 hexamer are listed in Tables S1 and S4.

The loss function for fitting the coarse-grained forces is

$$L_{\mathrm{inst}} = \sum_{t=1}^{N_{\mathrm{t}}} \left[ \sum_{I=1}^{N} \left( \alpha \left| \boldsymbol{F}_{\mathrm{FG},I}(\boldsymbol{r}_t^n) + \frac{\partial U(\xi_t)}{\partial R_I} \right|^2 + \beta \left| \boldsymbol{\tau}_{\mathrm{FG},I}(\boldsymbol{r}_t^n) + \sum_q \Omega_{I,q}(\xi_t)) \times \frac{\partial U(\xi_t))}{\partial \Omega_{I,q}} \right|^2 \right) \right], \tag{S3}$$

a modified version of the one used in Chapter 3, where the virial matching has been removed since simulations were done at constant volume instead of constant pressure. Here, $N_{\mathrm{t}}$ is the number of simulation configurations in the dataset, $\boldsymbol{r}_t^n$ are the fine-grained coordinates for configuration $t$, and $\xi_t = (\boldsymbol{R}^N(\boldsymbol{r}_t^n), \boldsymbol{\Omega}^N(\boldsymbol{r}_t^n))$ is the mapped coarse-grained configuration for this fine-grained configuration. The loss function is optimized using the minibatch gradient descent as implemented in TensorFlow. Where $\alpha, \beta$, and $\gamma$ are weights which specify the fraction of each loss that is used for backpropagation and were free to change with the learning rate during optimization

## SII. SEXITHIOPHENE COARSE-GRAIN POTENTIAL PARAMETERS

TABLE S1: Hyperparameters for sexithiophene single-site model

| hyperparameter | value | units |
|---|---|---|
| $\lambda$ | [-1.0, 1.0] | |
| $\eta$ | [2.0, 1.0] | $\mathring{A}^{-2}$ |
| $\nu$ | [2.0, 4.0, 8.0, 16.0, 32.0, 64.0] | |
| $R_\mathrm{s}$ | [0.5, 2.7, 5.0, 7.3, 9.6, 11.8, 14.2, 16.4, 18.7, 21.0] | Å |
| $R_\mathrm{c}$ | [ 21.0] | Å |

TABLE S2: Hyperparameters for sexithiophene six-site model

| hyperparameter | value | units |
|---|---|---|
| $\lambda$ | [-1.0, 1.0] | |
| $\eta$ | [2.0, 1.0] | $\mathring{A}^{-2}$ |
| $\nu$ | [2.0, 4.0, 8.0, 16.0, 32.0, 64.0] | |
| $R_\mathrm{s}$ | [0.5, 2.7, 5.0, 7.3, 9.6, 11.8, 14.2, 16.4, 18.7, 21.0] | Å |
| $R_\mathrm{c}$ | [ 21.0] | Å |

TABLE S3: Sexithiophene six-site model bond, angle, and dihedral parameters

| Parameter | value | Units |
|---|---|---|
| $K_\mathrm{B}$ | 200 | kcal/mol.Å |
| $b_0$ | 0.2 | Å |
| $K_\mathrm{D}$ | 6.45 | |
| $K_\mathrm{A}$ | 14.52 | kcal/mol/rad$^2$ |
| $\theta_0$ | 180 | (°) |

The dihedral parameters $K_1$, $K_3$, $K_4$ were all set to 0.

## SIII. N2200 COARSE-GRAIN POTENTIAL PARAMETERS

TABLE S4: Hyperparameters for N2200 hexamer six-site model

| hyperparameter | value | units |
|---|---|---|
| $\lambda$ | [-1.0, 1.0] | |
| $\eta$ | [2.0, 1.0] | $\text{Å}^{-2}$ |
| $\nu$ | [2.0, 4.0, 8.0, 16.0, 32.0, 64.0] | |
| $R_\mathrm{s}$ | [0.5, 1.3, 2.5, 4.7, 8.0, 10.3, 15.6, 21.8, 26.2, 30.4, 37.7, 42.0] | Å |
| $R_\mathrm{c}$ | [ 43.0] | Å |

TABLE S5: N2200 hexamer six-site model bond, angle, and dihedral parameters.

| Parameter | value | |
|---|---|---|
| $K_\mathrm{B}$ | 200 | kcal/mol.Å |
| $b_0$ | 0.2 | Å |
| $K_\mathrm{D}$ | 13.45 | |
| $K_\mathrm{A}$ | 29.52 | kcal/mol/rad$^2$ |
| $\theta_0$ | 180 | (°) |

The dihedral parameters $K_1$, $K_3$, $K_4$ were all set to 0.

# Statement of Authorship

| | |
|---|---|
| Title of Paper | Automatic labeling and prediction of anisotropic semiflexible polymers aggregate phase diagrams using neural networks |
| Publication Status | □ Published □ Accepted for Publication<br>□ Submitted for Publication ☑ Unpublished and Unsubmitted work written in manuscript style |
| Publication Details | |

## Principal Author

| | | | |
|---|---|---|---|
| Name of Principal Author (Candidate) | Marltan O. Wilson | | |
| Contribution to the Paper | Designed and trained machine learning algorithms, carried out simulations,<br>analyze and interpret the results, and compose manuscript. | | |
| Overall percentage (%) | | | |
| Certification: | This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper. | | |
| Signature | | Date | 15/12/2022 |

## Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

| | | | |
|---|---|---|---|
| Name of Co-Author | David Huang | | |
| Contribution to the Paper | Project conceptualization, research supervision, data analysis, writing (review and editing) | | |
| Signature | | Date | 09/01/2023 |

| | | | |
|---|---|---|---|
| Name of Co-Author | | | |
| Contribution to the Paper | | | |
| Signature | | Date | |

Please cut and paste additional co-author panels here as required.

# Automatic labeling and prediction of anisotropic semiflexible polymers aggregate phase diagrams using neural networks

Marltan O. Wilson[1] and David M. Huang[1]

*Department of Chemistry, School of Physics, Chemistry and Earth Sciences,*
*The University of Adelaide, Adelaide, South Australia 5005, Australia*

(*Electronic mail: david.huang@adelaide.edu.au)

(Dated: 10 January 2023)

A machine learning pipeline has been developed to understand the role of polymer backbone flexibility in the temperature-dependent A machine learning pipeline has been developed to understand the role of polymer backbone flexibility in the temperature-dependent aggregation behavior of anisotropic polymers. A toy polymer model is used to conduct simulations with variations in a predefined set of polymer properties. The set of variable properties used to model polymer backbone flexibility includes the coefficient of the angle potential and the coefficient of the dihedral potential. The temperature of the simulation is also used as a variable to determine the effect of temperature on the polymer conformations observed. The machine-learning pipeline developed was able to assign an aggregate type to unlabelled polymer trajectories as well as predict the type of aggregate based on the predefined properties of the polymer interaction potential.

## I. INTRODUCTION:

Organic semiconducting polymers, which typically consist of highly anisotropic monomers, are a major area of focus in the search for cheap, flexible, and printable optoelectronic devices such as light-emitting diodes and photovoltaic cells.[1–3] The ability to tune the polymer's flexibility and solubility makes them ideal for solution processing.[4–6] However, to maximize charge transport and overall device efficiency, a deeper understanding of the polymer aggregation process and the drivers of this process is needed.[7] The charge transport capabilities of an organic semiconductor are affected by chain size, persistent length, and overall crystallinity of the polymer.[8] mesoscopic features such as crystallinity and grain sizes are further driven by molecular properties such as the dihedral angle between monomers and processing conditions such as temperature and annealing rates[9]. To fully conceptualize the design space of organic semiconducting devices, mesoscopic polymer aggregation predictions must be able to consider both molecular properties and processing conditions.

Computational approaches such as molecular dynamics simulations play an important role in bridging the atomistic and mesoscopic length scales.[10] However, atomistic simulations of bulk polymer aggregates on equilibrium time scales are not feasible. To bridge the gap between atomistic and mesoscopic time scales, coarse-grained (CG) simulations are often used.[11] It is however important to note that, anisotropic polymers are best represented by anisotropic subunits capable of capturing the $\pi$–$\pi$ stacking configuration between polymers using a single CG site.[12] To this end, a significant amount of research has gone into the development of anisotropic potentials and coarse-grained models[13–16] capable of reproducing the bonds, angles, and dihedral distributions of the polymer backbone and side chains. These CG models allow for the efficient sampling of the conformational space of anisotropic polymers by tuning the backbone flexibility.

The conformational space of polymer organic semiconductors is a high-dimensional space with highly complex relationships between parameters. Machine learning has been effective in processing data from high-dimensional data sets while providing useful insight into the complex relationship between input and target variables.[17] There have been significant advances in the accessibility of machine learning to design powerful architectures with off-the-shelf layers and functions.[18] It is especially easy to design variational autoencoders for dimensionality reduction problems and feed-forward classification networks which are useful in grouping large amounts of data into predefined disjoint sets.[19] There have been previous attempts at using non-machine learning approaches to predict the aggregation behavior of semiflexible polymers with strictly isotropic monomers.[20] Previous works, also explored the aggregation phase diagram of semiflexible polymers using molecular dynamics simulations without predictive capabilities.[21] Machine-learning approaches have been explored with great success, especially in the field of computational biology.[22]

In this work, we develop two data-driven workflows assisted by machine learning to identify, classify and predict the types of polymer aggregates obtained from simulating anisotropic polymers with varying properties under different simulation conditions. The first algorithm uses an autoencoder to subdivide the entire conformational space of the simulated polymer into a predefined number of disjoint sets that can be easily labeled manually. The second algorithm attempts to predict the most probable polymer aggregate to form under specific simulation conditions for a given set of molecular scale polymer properties. Together, these algorithms are capable of combining molecular scale properties and processing conditions to predict the mesoscopic bulk behavior of polymer aggregates and potentially inform design choices for organic optoelectronic devices.

## II. THEORY

### A. Anisotropic polymer model

The generalized coarse-grained polymer model and procedure used for the simulations have been fully described in previous works.[23] These coarse-grained polymers have been designed with the Gay-Berne biaxial potential for dissimilar particles[24,25] and explicit inclusion of dihedral angles between nearest-neighbor anisotropic monomers. The anisotropic Gay-Berne potential is implemented in the LAMMPS package[26] and is given by the expression[12]

$$U_{\mathrm{GB}}(\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{r}_{12}) = U_r(\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{r}_{12},\gamma)\cdot\eta_{12}(\boldsymbol{A}_1,\boldsymbol{A}_2,\nu)\cdot \chi_{12}(\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{r}_{12},\mu) \quad (1)$$

where

$$U_r = 4\varepsilon(\rho^{12}-\rho^6) \quad (2)$$

$$\rho = \frac{\sigma}{h_{12}+\gamma\sigma} \quad (3)$$

where $\boldsymbol{r}_{ij}$ is the distance between the centers-of-mass of the two ellipsoids, $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$ are the rotation matrices transforming the orientation of the ellipsoids from lab frame to body frame. $h_{12}$ is the approximation to the distance of closest approach and $\gamma$ and $\mu$ are both set to 1.0. Reduced LJ units are used, so lengths are in units of $\sigma$, energy in units of $\varepsilon$ and temperature in units of $\varepsilon/k_{\mathrm{B}}T$ . The mass is in units of the monomer mass $m$ and time is in units of $\sqrt{m\sigma^2/\varepsilon}$.[12] Each monomer has noninteracting "ghost" atoms attached at off-center positions for the definition of bonds between ellipsoids. This ensures that forces and torques are correctly applied to the anisotropic particle and not just the center of mass of the monomer. The polymer semiflexibility and dihedral barrier height are determined by the following equations for the bond length, bond angle, and dihedral angle potentials,[12]

$$E_{\mathrm{bond}} = K_{\mathrm{B}}(b-b_0)^2, \quad (4)$$

$$E_{\mathrm{angle}} = K_{\mathrm{A}}(\theta-\theta_0)^2, \quad (5)$$

$$E_{\mathrm{dihedral}} = \frac{1}{2}K_1[1+cos(\phi)]+\frac{1}{2}K_{\mathrm{D}}[1-cos(2\phi)] +\frac{1}{2}K_3[1+cos(3\phi)]+\frac{1}{2}K_4[1-cos(4\phi)], \quad (6)$$

where $b$ and $b_0$ are the instantantous and equilibrium bond lengths, respectively, $\theta$ and $\theta_0$ are the instantaneous and equilibrium bond angles, respectively, $\phi$ is the dihedral angle, $K_{\mathrm{B}}$ and $K_{\mathrm{A}}$ are the bond and three-body angle potential parameter, respectively, and $K_1$, $K_{\mathrm{D}}$, $K_3$, and $K_4$ are the coefficient of the OPLS cosine expansion of the dihedral potential.

The angle coefficient $K_{\mathrm{A}}$ and the second coefficient of the OPLS cosine expansion $K_{\mathrm{D}}$ are manipulated to represent various backbone flexibility of typical organic semiconductors. For this work the length of the polymer chain was varied between 22 and 64 monomers and the other coarse-grained we selected was in line with previously published results.[12] A schematic of the bonding and the definition of the dihedral angle is shown in Fig. 1.

### B. Neural network architecture

To use a machine learning approach to construct a phase space of aggregates parameterized by the polymer molecular properties and processing conditions, there has to be a systematic approach to the identification and classification of the polymer aggregates obtained from long simulations. A variational autoencoder[27] implementation is ideal for the unsupervised labeling of all configurations obtained from simulations. This variational autoencoder shown in Fig. 2 is constructed from an encoder network and a decoder network.[28] The encoder maps a set of inputs to a mean $\mu$ and standard deviation $\sigma$. It then samples from the standard normal distribution to create the latent space $\boldsymbol{Z}$.[29,30] Using a variational autoencoder that samples from a normal distribution ensures that the latent space can be interpolated. The latent space $\boldsymbol{Z}$ can then be divided into disjoint sets by resampling from a relaxed one-hot categorical distribution before reconstructing it with a decoder network. The Gumbel-softmax reparameterization trick is used to approximate an argmax function through the introduction of a neural network temperature variable.[31,32] Once determined, these disjoint sets represent the labels of different aggregates found in the training data set. During training, the neural network temperature variable is gradually reduced to anneal each configuration into a unique aggregate label. The decoder network takes the output of the encoder as an input and tries to reconstruct the input parameters of the encoder from the latent space representation. The loss function of the autoencoder is calculated as a reconstruction and regularization loss,[4] where the reconstruction error minimizes the difference between the input of the encoder and the output of the decoder and the regularization loss[33] attempts to minimize the distance between the true distribution and the distribution being sampled. This approach, where the input and the output of a feedforward neural network are the same, is considered unsupervised learning. This unsupervised learning approach reduces the prior knowledge about the polymer aggregation that is needed to find a set of the most distinct probable aggregates.

### C. Aggregate preprocessing

To optimize the neural network training, the polymer conformations obtained from the simulation have to be preprocessed into a representation that is invariant under translation and global rotation. Polymer configurations are first mapped to a spatial correlation matrix $\boldsymbol{M}$.[34] The $(i,j)$ element of the

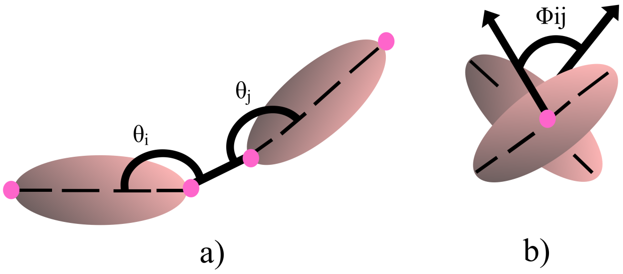


FIG. 1. Schematic of anisotropic polymer used for the simulations, showing, (a) the bond angle between anisotropic monomers defined using off-centered sites and (b) the dihedral angle between adjacent monomers.

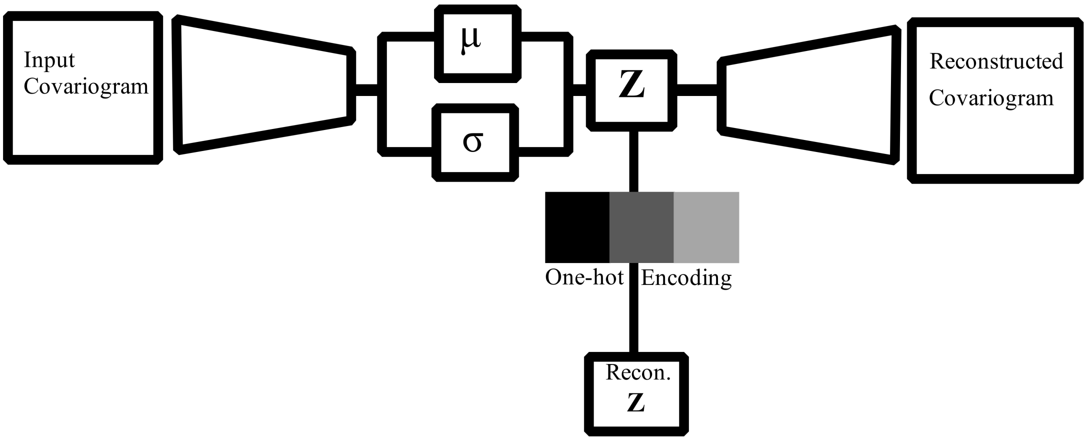


FIG. 2. Schematic of autoencoder

matrix is given by

$$M_{ij} = \boldsymbol{u}_i \cdot \boldsymbol{u}_j \tag{7}$$

where, $\boldsymbol{u}_i$ is the unit vector pointing from the center-of-mass of ghost atom $i$ to the center-of-mass of the $i+1$ ghost atom. This ensures that for an uncollapsed (open) polymer (Fig. 3c) $M_{ii} \equiv 1\ \forall i$ and decreases exponentially along the length of the chain for all values of $M_{ij}$. Hairpin-shaped aggregates (hairpins) (Fig. 3g) will display a square wave pattern with a flat area close to 1 corresponding to the first arm followed by an area of rapid decay to -1 corresponding to the head and finally a flat region at -1 corresponding to the second arm going in the opposite direction. Toroidal-shaped aggregates (toroids) (Fig. 3e) will present with a repeating sine wave corresponding to the number of loops making up the toroid. There are no flat regions in the toroid's spatial correlation matrix because it does not possess long arms such as those seen in hairpins. A further comparison of aggregate conformation, and the corresponding spatial correlation matrix and covariogram is shown in Fig. 4 The 2D spatial correlation matrix is then condensed into a 1D spatial covariogram, which acts as a statistical measure of the spatial covariance as a function of distance and is calculated as[34]

$$C(h) = \frac{1}{n(h)} \sum_{j=1}^{n} \sum_{i=1}^{n} (\boldsymbol{u}_i - \boldsymbol{\mu}) \cdot (\boldsymbol{u}_j - \boldsymbol{\mu}), \tag{8}$$

where $h$ is the distance in space between observation $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$, $n(h)$ is the number of observations at a distance $h$, and in this case $\boldsymbol{\mu} = \vec{0}$ is the mean. $C(h)$ is a scalar function bounded between 1 and -1. The values of $h$ are chosen from the range 0 to the length of the polymer $(L)$ The cardinality of the set is fixed for all polymers and is independent of the degree of polymerization. An exponential decay corresponds to an open polymer configuration. $C(h)$ for other configurations such as multi-head rackets and toroids oscillate between 1 and -1 and the number of zero crossings corresponds to the number of heads or loops.

In the case where the number of monomers differs between polymers, the spatial covariogram will also have different length vectors. To standardize the length of the covariogram vector, the number of elements is set to 63, and where the number of monomers is less than or greater than 63, the points are interpolated using cubic spline and then 63 new points are generated along the path.

Training of the aggregate labeling neural network using a

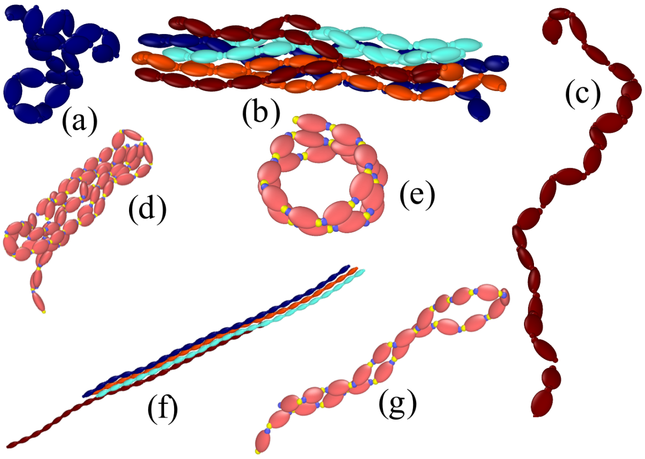


FIG. 3. Typical aggregates in simulation: (a) orientationally disordered globule, (b) flexible four chain bundle, (c) open, (d) multi-head racket, (e) toroid, (f) rigid four chain bundle and (g) hairpin

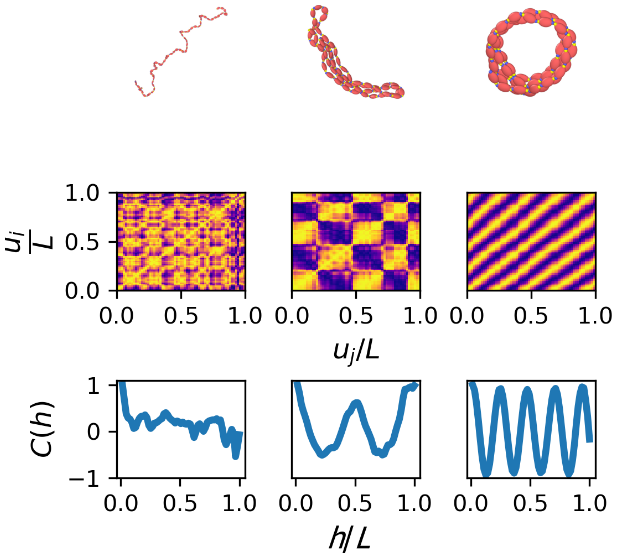


FIG. 4. Typical aggregates along with the corresponding heatmap of the spatial correlation matrix and the spatial covariogram.

standardized data set requires the generation of data representative of possible aggregate types that would be observed in the coarse-grained polymer simulations. To generate this training data set, the list of aggregate classes considered is as follows:

1. Open is used to describe any polymer that has not collapsed into an aggregate.
2. Hairpin describes hairpin-shaped aggregates.
3. Multi-head racket describes an aggregate with more than one racket-shaped head.
4. Toroidal is used to describe all looped polymers independent of the shape or the number of loops.

A standard dataset for each of these classes of aggregates was created by manually selecting examples of the spatial covariogram associated with each of the aggregate types from the available training data and adding noise to make the training of the neural network more robust. This standardized data set ensured that all aggregates in the coarse-grained simulations were compared to and mapped to one of the possible aggregate types above. However, when the self-referential route was taken, the labeling autoencoder was trained on the spatial covariogram obtained from the simulated polymer trajectories. The training dataset obtained from molecular dynamics simulations was unbalanced due to the difference in the lifetime of various aggregates. To account for this variation in the training data, the autoencoder was trained iteratively. On the first run, a random batch of 50,000 polymer configurations was used to train the autoencoder. In each subsequent run, the trained neural network was used to evaluate the full set of available training data then the subset of data used for training was increased by 10% by adding in the polymer configurations with the largest error. The actual training data was then evaluated using the trained neural network and the bottom 1% with the smallest error was removed from the training subset. The iterative updating of the training subset was done until the average error of the training subset was equivalent to the average error of the available training data. This iterative method ensured that overrepresented configurations in the training subset were removed and rare ones were added. The autoencoder was trained on a subset of 100,000 data points from the available $4 \times 10^6$ unique polymer trajectories. The benefit of the self-referential approach over the standardized data set was that new types of aggregates can be discovered and the latent space consisted of the most probable types of aggregates. There were however some disadvantages compared to the standardized dataset. The most significant was that the aggregate classes of the latent space have to be manually labeled after the training of the neural network was completed.

The same procedure was used to determine the conformation of each polymer in a multichain aggregate. However, a further spatial metric shown in Fig. 5 was defined between each polymer to determine the degree of overlap between the monomers of each polymer, the matrix $\mathbf{\Delta}^{IJ}$, whose $(i,j)$ element is

$$\Delta_{ij}^{IJ} = -\tanh\left(\frac{\|\mathbf{r}_{I,i} - \mathbf{r}_{J,j}\|}{\alpha\sigma}\right) + 1 \tag{9}$$

where $r_{I,i}$ is the position of monomer $i$ of polymer $I$, and $r_{J,j}$ is the position of monomer $j$ of polymer $J$ and $\sigma$ is the same as Eqn. (3) and $\alpha$ is an integer to scale the aggregation cut-off distance. While the covariogram describes the conformation of each polymer, the matrix $\mathbf{\Delta}^{IJ}$ describes the degree of overlap between polymers and highlights the position along the polymer with the highest interchain aggregation.

### D. Aggregate prediction

Once all the aggregates from the molecular dynamics simulations have been labeled they could be used to predict the most probable aggregate that would occur under different conditions for a given set of polymer features. A machine learning approach allowed for the creation of high-dimensional aggregate phase diagrams. In this case, the set of polymer parameters along with the simulation condition was used as input $\mathscr{D}$ to the neural network shown in Fig. 6, i.e.

$$\mathscr{D} = [\tau, K_{\mathrm{A}}, K_{\mathrm{D}}, N, T], \tag{10}$$

where $K_{\mathrm{A}}$ and $K_{\mathrm{D}}$ are the angle and dihedral coefficient, $T$ and $N$ are the simulation temperature and degree of polymerization, and $\tau$ is the time-like variable since parallel tempering was used in the simulation, but the same analysis could be used for simulation trajectories with unbiased dynamics to predict non-equilibrium phase diagrams. All the parameters were scaled between 0 and 1 since their raw values had orders of magnitude differences.

The output of the aggregate prediction network was then compared to the labels obtained from the labeling autoencoder. The aggregate prediction neural network could explore the aggregate phase space of the polymer and visually inspect how phase boundaries change over time or with temperature and flexibility.

### E. Molecular Dynamics

Molecular dynamics simulations were performed using the LAMMPS package with modification to include an explicit anisotropic dihedral potential, a list of the corresponding parameters for the interaction potential can be found in the Supplementary Material. An implicit solvent model was used where the solvent was incorporated via renormalization of the intermolecular interactions and the use of the Langevin thermostat. Langevin simulations used a damping parameter of 2 and a timestep of 0.00075. Simulations were performed for chain lengths between 22 and 64 monomers in a volume of 100 ($\sigma^3$) and the number of chains in each simulation varied between 1 and 8. The polymer simulations were performed using parallel tempering. The temperature spacings between replicas are adjusted such that an acceptance ratio of 20–30 % is achieved for all replicas. This was used to sample a wide variety of temperatures and the complete configurational space of the polymer aggregation.

25 different simulations were done for different combinations of $K_{\mathrm{A}}$ and $K_{\mathrm{D}}$. The value of the $K_{\mathrm{A}}$ parameter was taken from the range $1 \leq K_{\mathrm{A}} \leq 5$, similarly the $K_{\mathrm{D}}$ parameter was set to a value in the range $1 \leq K_{\mathrm{D}} \leq 5$. Each simulation was done using parallel tempering with the temperature range of $0.1 \leq T \leq 1.5$ for a total of 250 different combinations of $K_{\mathrm{A}}$, $K_{\mathrm{D}}$, and $T$. Different types of polymer aggregates were observed based on the chain length and flexibility, temperature, number of chains, and the length of the simulation. The aggregates ranged from orientationally disordered globules of single chains at low temperatures and high flexibility to open rod-like multi-chain aggregates at high temperatures and low flexibility. Plots of the spatial correlation matrix and the spatial covariogram are obtained by analyzing trajectories from the simulation data. This set of known aggregate types acts as a reference key for manually assigning a name to the significant aggregate labels obtained from the encoder latent space.

The latent space of the autoencoder was set to 8 disjoint sets, to obtain eight unique aggregate labels and the neural network temperature variable was set to 2.

The temperature in the Gumbel distribution was gradually reduced by 1% each epoch until it reaches a value of 0.01. The fraction of each aggregate class was then obtained from the ratio of the number of aggregates assigned to each class to the total number of aggregates in the simulation data set.

## III. RESULTS AND DISCUSSION

Training of the autoencoding neural network produced the latent space, which can be visualized as a linear sequence of polymer aggregates parameterized by a single value, as shown in Fig. 7. The latent space was constructed such that aggregates with similar covariograms were grouped close to each other, ensuring smoother transitions in the phase space representation once the aggregates are given a unique aggregate label. By assigning each polymer trajectory to a unique aggregate label, the relative proportion of each aggregate in the data set could be determined. Fig. 8 shows the expected unbalanced dataset where the aggregate labeled A3 accounts for close to 70% of all observed aggregates.

The one-hot vector associated with the aggregate label A3 could then be passed to the decoder to find the corresponding covariogram from which the general structure of the aggregate was determined. Therefore, the decoder portion of the autoencoder must have high reconstruction fidelity. The reconstruction fidelity of the decoder can be evaluated by comparing the

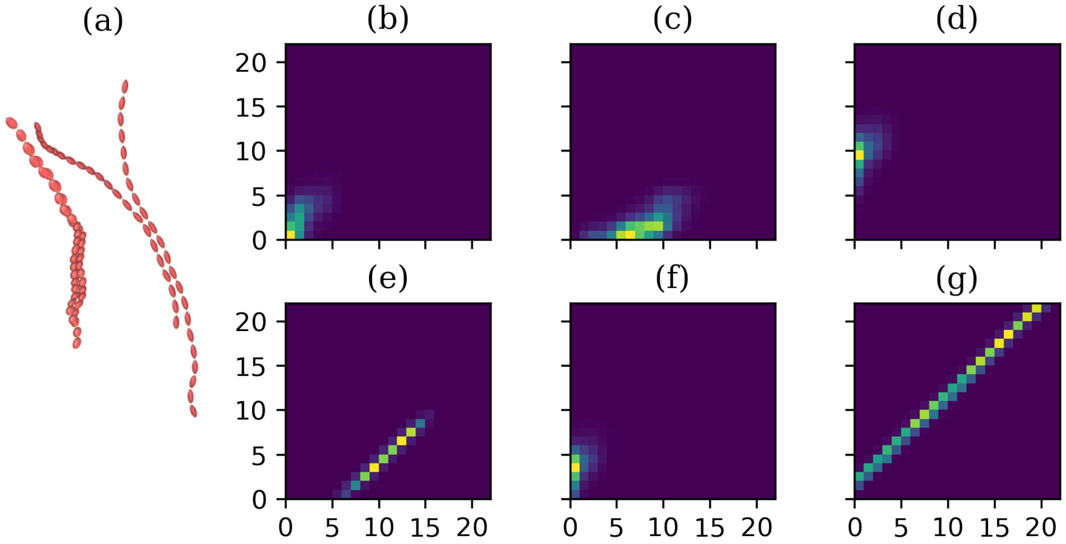


FIG. 5. The six combinations of the $\mathbf{\Delta}^{IJ}$ matrix for a four polymer system (a). The partially aggregated system shows a strong alignment between two pairs of polymers in (e) and (g)

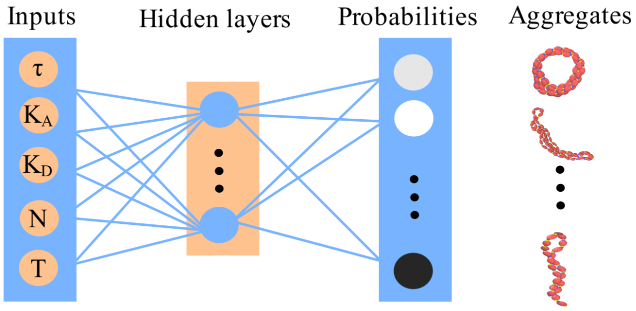


FIG. 6. Schematic of classifier neural network with inputs defined in Eqn. (10)

true and the reconstructed covariogram of data that the neural network did not use for training. The decoder portion of the autoencoder could reconstruct random selections of aggregates taken from the test set as shown in Fig. 9. Even with a 63:1 compression, the general shape of the covariogram was preserved with only minor deviations where the covariogram was noisy.

It is expected that the conditions under which a polymer is simulated along with its intrinsic properties should determine the types of aggregates produced. When vector $\mathscr{D}$ was used as input to predict the corresponding latent variable derived from the autoencoding network, the trained classifier network could construct the expected phase space of any polymer, which lies in the range spanned by the simulated polymer trajectories. Snapshots of the neural network predicted phase space are presented in Figs. 10–12. Each slice of the high-dimensional phase space plot the aggregate latent space parameter as a function of elements of vector $\mathscr{D}$.

From the plots of the phase diagram, it could be determined that the open polymer dominated at high temperatures and regions where the polymer was relatively stiff. The neural network aggregate phase model also showed that flexible long-chain polymers at lower temperatures Fig. 10a formed a more coiled aggregate while short-chain polymers at the same temperature (Fig. 10c) were less likely to do so. For small values of $K_\text{A}$ the anisotropic polymers are expected to aggregate at all temperatures and for all chain lengths. However, for longer chains and lower temperatures, the anisotropic polymer systems form toroids with multiple loops, as shown in Fig. 11. These toroidal aggregates are only expected to form for small values of $K_\text{A}$ and $K_\text{A}$ at low temperatures, as shown in Fig. 12. The length of the polymer chain also played a significant role in determining if toroidal aggregates are formed, since they are less probable for short-chained polymers shown in Fig. 12c The changes related to the effect of time on the long-chain semiflexible polymer are shown in Fig. 13, from partial collapse at small $\tau$ to the equilibrium aggregate structure at large $\tau$.

There are similarities between the neural network predicted equilibrium phase diagram shown in Fig. 13c and previously published results for the same set of polymer parameters and simulation conditions.[12] Both the previously published simulated phase diagram[12] and the predicted phase diagram in Fig. 13c show the open polymer to be the most abundant conformation at large temperature values $(T > 0.7)$ while aggregates with multiple loops were abundant at low temperatures $(T < 0.2)$. These multi-loop aggregates were independent of the value of $K_\text{A}$ at low temperatures but as temperature increased, there was a transition to a single-loop aggregate at intermediate temperatures similar to previous results,[12] which showed racket-shaped aggregates as the most common at intermediate temperatures, However, the neural network pre-

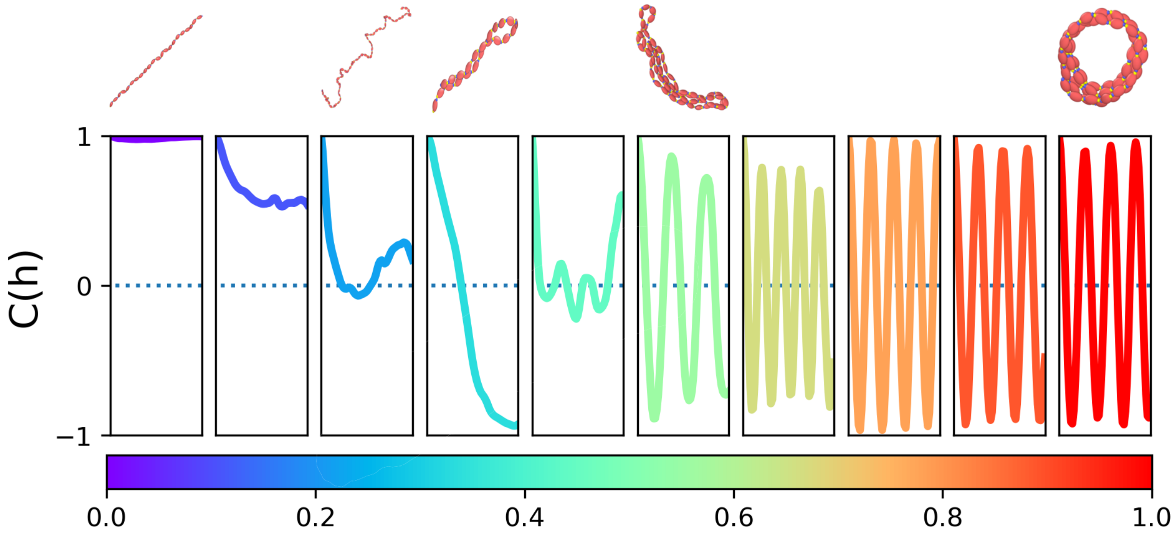


FIG. 7. The reconstructed covariograms derived from a sequence of linearly spaced values in the latent space of the autoencoder. The color of each covariogram corresponds to the value of its latent space representation.

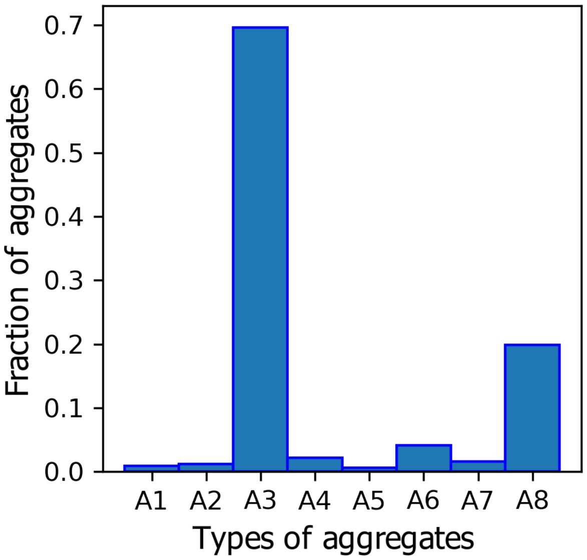


FIG. 8. The relative fraction of each observed aggregate in the available training dataset.

dicted phase diagram in Fig. 13(c) does not show a distinct transition region at $1 < K_A < 2$.

There are similarities between the phase diagrams of isotropic polymers[21] and the neural network predicted phase diagrams shown in Fig. 10, especially with respect to the aggregate dependence on temperature, but the transitions between aggregate phases largely happen at different temperatures and stiffness when comparing isotropic and anisotropic polymers as shown in Fig. 14, multi-chain aggregation of the anisotropic polymers was also similar to the aggregation of

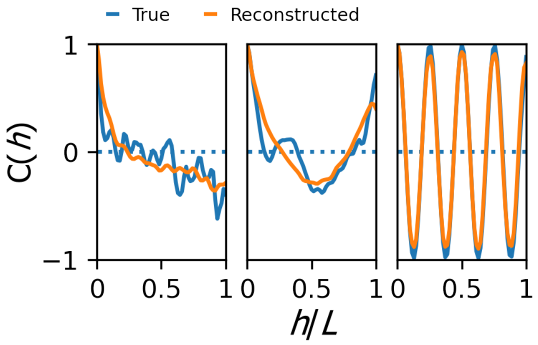


FIG. 9. Examples of reconstructed covariograms corresponding to a (left) hairpin, (middle) multi-headed racket, and (right) toroid

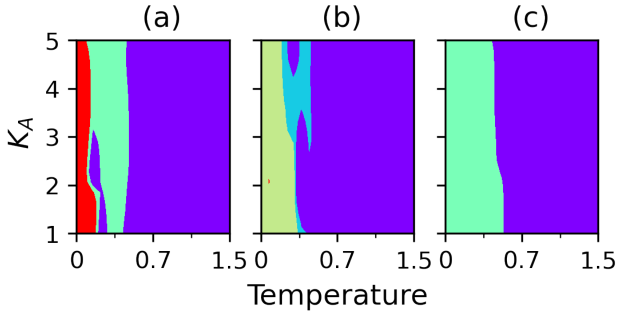


FIG. 10. $K_A$ versus temperature slices from the high-dimensional phase diagram of anisotropic polymer at equilibrium with $K_D = 1$, for (a) 64-, (b) 44-, and (c) 22-monomer chain. The color bar and associated covariogram are shown in Fig. 7

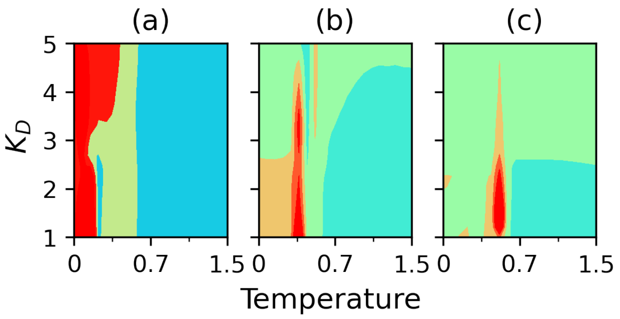


FIG. 11. $K_{\mathrm{D}}$ versus temperature slices from the high-dimensional phase diagram of anisotropic polymer at equilibrium with $K_{\mathrm{A}} = 1$, for (a) 64- (b) 44- (c) 22-monomer chain. The color bar and associated covariogram are shown in Fig. 7

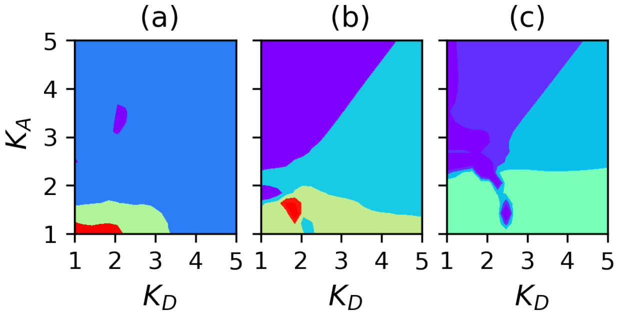


FIG. 12. $K_{\mathrm{A}}$ versus $K_{\mathrm{D}}$ slices from the high-dimensional phase diagram of anisotropic polymer at equilibrium with temperature = 0.3, for (a) 64- (b) 44- (c) 22-monomer chain. The color bar and associated covariogram are shown in Fig. 7.

multi-chain isotropic polymers[21]. Each polymer in the pair of aggregated polymers shown in Fig. 14a was predicted to form a single racket-shaped aggregate at low temperatures ($T$<0.35), while Fig. 14b showed that the hairpins were interlocked. At higher temperatures ($T > 4$), the neural network predicts an open configuration, and Fig. 14c shows that the pair are expected to completely overlap to form a rod-like aggregate. Additional phase diagrams can be found in the Supplementary Material.

## CONCLUSIONS

An unsupervised aggregate labeling autoencoder neural network was developed to assign an aggregate type to trajectories from large simulations either by comparison to a standard set of aggregates or by a self-referential route. We further showed that this labeled data can be used alongside the polymer molecular scale parameters and the simulation conditions, to predict the most likely polymer aggregates to occur under different processing conditions, polymer flexibility, and degree of polymerization. The results confirm that there is a strong correlation between the molecular scale parameters, the processing conditions, and the equilibrium conformation of anisotropic polymer semiconductors. The neural network method was able to predict that the number of loops formed from a single chain aggregate decreases with temperature. Toroidal aggregates are also more abundant for small values of $K_{\mathrm{A}}$ and $K_{\mathrm{D}}$ (<2). For multi-chain aggregation, the rod-like structure was most common at equilibrium except for highly flexible polymers at low temperatures which formed interlocking hairpins. By comparing slices from the neural network constructed phase diagrams we have shown there is good agreement with previously published results using the same polymer systems. This machine learning approach, trained on coarse-grained simulations has the potential to reduce the number of atomistic simulations and experiments needed to explore the aggregate phase space when designing organic semiconductor devices. The accuracy for specific polymer systems can be further increased through top-down fine-tuning of the polymer interaction potentials and dynamics.

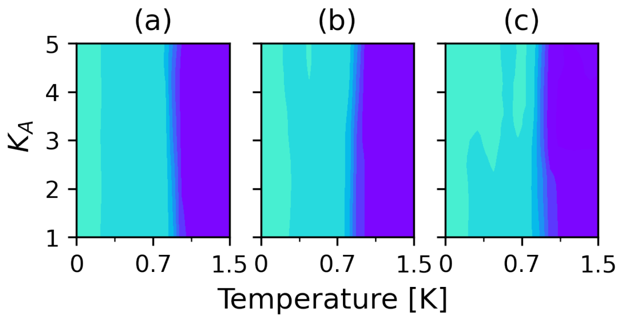


FIG. 13. $K_{\mathrm{A}}$ versus temperature slices from the high-dimensional phase diagram of anisotropic polymer with $K_{\mathrm{D}} = 3$ for (a) small, (b) medium, and (c) large $\tau$. The color bar and associated covariogram are shown in Fig. 7

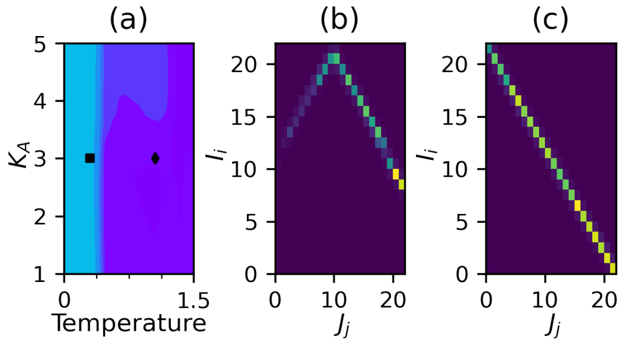


FIG. 14. (a) $K_{\mathrm{A}}$ versus temperature slice from the high-dimensional phase diagram of an aggregated pair of anisotropic polymer at equilibrium with $K_{\mathrm{D}} = 3$ (color bar and associated covariogram are shown in Fig. 7). (b) The $\mathbf{\Delta}^{IJ}$ matrix for the pair of aggregated polymer with $K_{\mathrm{D}} = 3$ and $T = 0.3$ and (c) the $\mathbf{\Delta}^{IJ}$ matrix for the pair of aggregated polymer with $K_{\mathrm{D}} = 3$ and $T = 1.0$.

## ACKNOWLEDGMENTS

This work was supported by the Australian Research Council under the Discovery Projects funding scheme (DP190102100). This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government, and from the University of Adelaide's Phoenix High-Performance Computing service.

## REFERENCES

[1] M. J. Han, D. Wei, H. S. Yun, S.-h. Lee, H. Ahn, D. M. Walba, T. J. Shin, and D. K. Yoon, "Precise orientation control of a liquid crystal organic semiconductor via anisotropic surface treatment," NPG Asia Mater. **14**, 1–12 (2022).

[2] J. Lee, S. A. Park, S. U. Ryu, D. Chung, T. Park, and S. Y. Son, "Green-solvent-processable organic semiconductors and future directions for advanced organic electronics," J. Mater. Chem. A **8**, 21455–21473 (2020).

[3] C. Yumusak, N. S. Sariciftci, and M. Irimia-Vladu, "Purity of organic semiconductors as a key factor for the performance of organic electronic devices," Mater. Chem. Front. **4**, 3678–3689 (2020).

[4] S. Wang, L. Peng, H. Sun, and W. Huang, "The future of solution processing toward organic semiconductor devices: a substrate and integration perspective," J. Mater. Chem.C **10**, 12468–12486 (2022).

[5] V. N. Hamanaka, E. Salsberg, F. J. Fonseca, and H. Aziz, "Investigating the influence of the solution-processing method on the morphological properties of organic semiconductor films and their impact on oled performance and lifetime," Org. Elect. **78**, 105509 (2020).

[6] S. Allard, M. Forster, B. Souharce, H. Thiem, and U. Scherf, "Organic semiconductors for solution-processable field-effect transistors (ofets)," Angewandte Chemie Int. Ed. **47**, 4070–4098 (2008).

[7] H. Hu, P. C. Chow, G. Zhang, T. Ma, J. Liu, G. Yang, and H. Yan, "Design of donor polymers with strong temperature-dependent aggregation property for efficient organic photovoltaics," Acc. Chem. Res. **50**, 2519–2528 (2017).

[8] S. Liu, W. M. Wang, A. L. Briseno, S. C. Mannsfeld, and Z. Bao, "Controlled deposition of crystalline organic semiconductors for field-effect-transistor applications," Adv. Mater. **21**, 1217–1232 (2009).

[9] K. C. Dickey, J. E. Anthony, and Y.-L. Loo, "Improving organic thin-film transistor performance through solvent-vapor annealing of solution-processable triethylsilylethynyl anthradithiophene," Adv. Mater. **18**, 1721–1726 (2006).

[10] M. Praprotnik, L. Delle Site, and K. Kremer, "Adaptive resolution scheme for efficient hybrid atomistic-mesoscale molecular dynamics simulations of dense liquids," Phys. Rev. E **73**, 066701 (2006).

[11] N. E. Jackson, "Coarse-graining organic semiconductors: the path to multiscale design," J. Phys. Chem. B **125**, 485–496 (2020).

[12] A. E. Cohen, N. E. Jackson, and J. J. De Pablo, "Anisotropic coarse-grained model for conjugated polymers: Investigations into solution morphologies," Macromol. **54**, 3780–3789 (2021).

[13] M. Babadi, R. Everaers, and M. Ejtehadi, "Coarse-grained interaction potentials for anisotropic molecules," J. Chem. Phys. **124**, 174708 (2006).

[14] M. Ricci, O. M. Roscioni, L. Querciagrossa, and C. Zannoni, "Molc. a reversible coarse grained approach using anisotropic beads for the modelling of organic functional materials," Phys. Chem. Chem. Phys. **21**, 26195–26211 (2019).

[15] A. F. Tillack, L. E. Johnson, B. E. Eichinger, and B. H. Robinson, "Systematic generation of anisotropic coarse-grained lennard-jones potentials and their application to ordered soft matter," J. Chem. Theory Comput. **12**, 4362–4374 (2016).

[16] F. Goujon, N. Martzel, A. Dequidt, B. Latour, S. Garruchet, J. Devémy, R. Blaak, É. Munch, and P. Malfreyt, "Backbone oriented anisotropic coarse grains for efficient simulations of polymers," J. Chem. Phys. **153**, 214901 (2020).

[17] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Sci. **349**, 255–260 (2015).

[18] S. Kim, H. Wimmer, and J. Kim, "Analysis of deep learning libraries: Keras, pytorch, and mxnet," in *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)* (IEEE, 2022) pp. 54–62.

[19] S. J. Wetzel, "Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders," Phys. Rev.E **96**, 022140 (2017).

[20] J. Zierenberg and W. Janke, "From amorphous aggregates to polymer bundles: The role of stiffness on structural phases in polymer aggregation," EPL **109**, 28002 (2015).

[21] J. Zierenberg, M. Marenz, and W. Janke, "Dilute semiflexible polymers with attraction: Collapse, folding and aggregation," Poly. **8**, 333 (2016).

[22] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, "Highly accurate protein structure prediction with alphafold," Nat. **596**, 583–589 (2021).

[23] A. E. Cohen, N. E. Jackson, and J. J. de Pablo, "Anisotropic coarse-grained model for conjugated polymers: Investigations into solution morphologies," Macromol. **54**, 3780–3789 (2021).

[24] R. Berardi, C. Fava, and C. Zannoni, "A gay–berne potential for dissimilar biaxial particles," Chem. Phys. Lett. **297**, 8–14 (1998).

[25] R. Berardi, C. Fava, and C. Zannoni, "A generalized gay-berne intermolecular potential for biaxial particles," Chem. Phys. Let. **236**, 462–468 (1995).

[26] 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).

[27] S. V. Kalinin, O. Dyck, S. Jesse, and M. Ziatdinov, "Exploring order parameters and dynamic processes in disordered systems via variational autoencoders," Sci. Adv. **7**, eabd5084 (2021).

[28] R. Batra, H. Dai, T. D. Huan, L. Chen, C. Kim, W. R. Gutekunst, L. Song, and R. Ramprasad, "Polymers for extreme conditions designed using syntax-directed variational autoencoders," Chem. Mater. **32**, 10489–10500 (2020).

[29] C. Doersch, "Tutorial on variational autoencoders," arXiv (2016).

[30] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps, "Towards visually explaining variational autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020) pp. 8642–8651.

[31] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," arXiv (2016).

[32] J. Chang, X. Zhang, Y. Guo, G. Meng, S. Xiang, and C. Pan, "Differentiable architecture search with ensemble gumbel-softmax," arXiv (2019).

[33] S. Odaibo, "Tutorial: Deriving the standard variational autoencoder (vae) loss function," arXiv (2019).

[34] A. Montesi, M. Pasquali, and F. MacKintosh, "Collapse of a semiflexible polymer in poor solvent," Phys. Rev. E **69**, 021916 (2004).

# Supplementary Material:
# Automatic labeling and prediction of anisotropic semiflexible polymers aggregate phase diagrams using neural networks

Marltan O. Wilson and David M. Huang

*Department of Chemistry, School of Physics, Chemistry and Earth Sciences,*

*The University of Adelaide, Adelaide, South Australia 5005, Australia*

## CONTENTS

## SI. INTERACTION POTENTIAL PARAMETERS

TABLE S1: Gay–Berne parameters.

| Parameter | value |
|---|---|
| $\gamma$ | 1 |
| $\upsilon$ | 1 |
| $\mu$ | 1 |
| cutoff | 3 |
| $\varepsilon$ | 1 |
| $\sigma$ | 1 |
| $\sigma_a$ | 1.0 |
| $\sigma_b$ | 0.7 |
| $\sigma_c$ | 0.4 |
| $\varepsilon_a$ | 0.25 |
| $\varepsilon_b$ | 1.0 |
| $\varepsilon_c$ | 0.1 |

TABLE S2: Bond parameters.

| Parameter | value |
|---|---|
| $K_\mathrm{B}$ | 200 |
| $b_0$ | 0.2 |

The $\theta_0$ parameter in the angle potential is set to 180 and the dihedral parameters $K_1$, $K_3$, $K_4$ are all set to 0.

## SII. ADDITIONAL PHASE DIAGRAMS

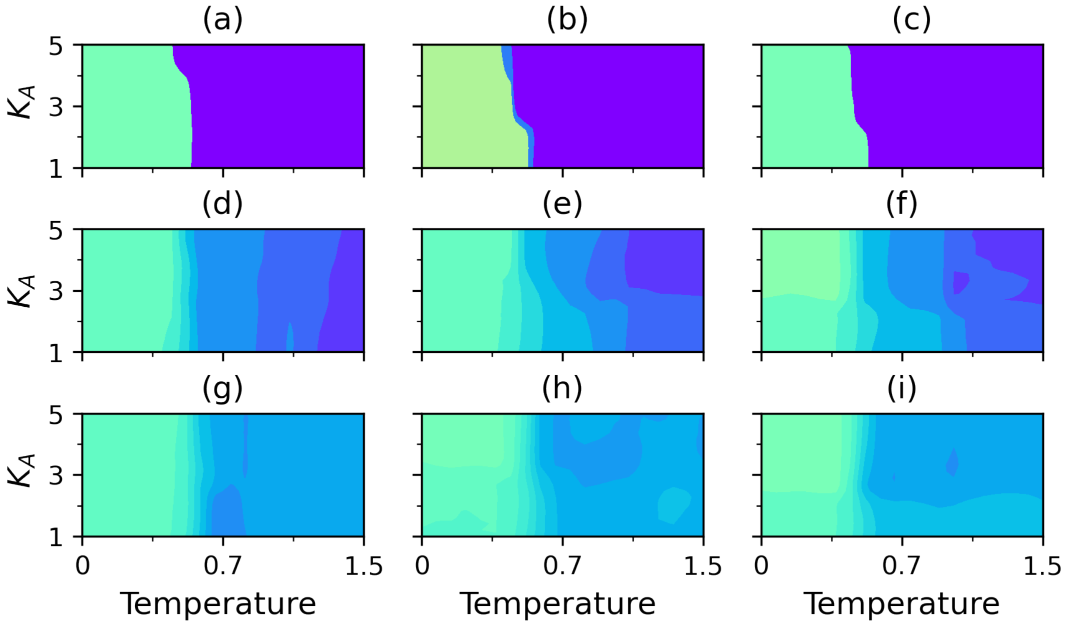


FIG. S1: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 22 monomers. Each plot shows the most probable polymer conformation as a function of temperature and chain semiflexibility parameter $K_A$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $K_D \in \{1, 3, 5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

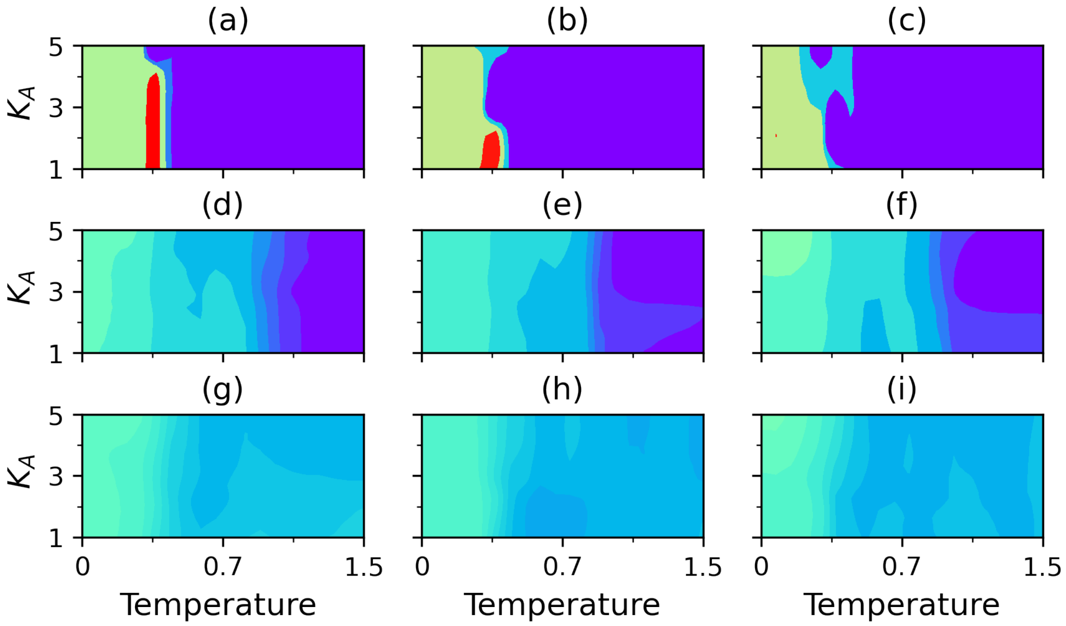


FIG. S2: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 44 monomers. Each plot shows the most probable polymer confirmation as a function of temperature and chain semiflexibility parameter $K_A$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $K_D \in \{1, 3, 5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

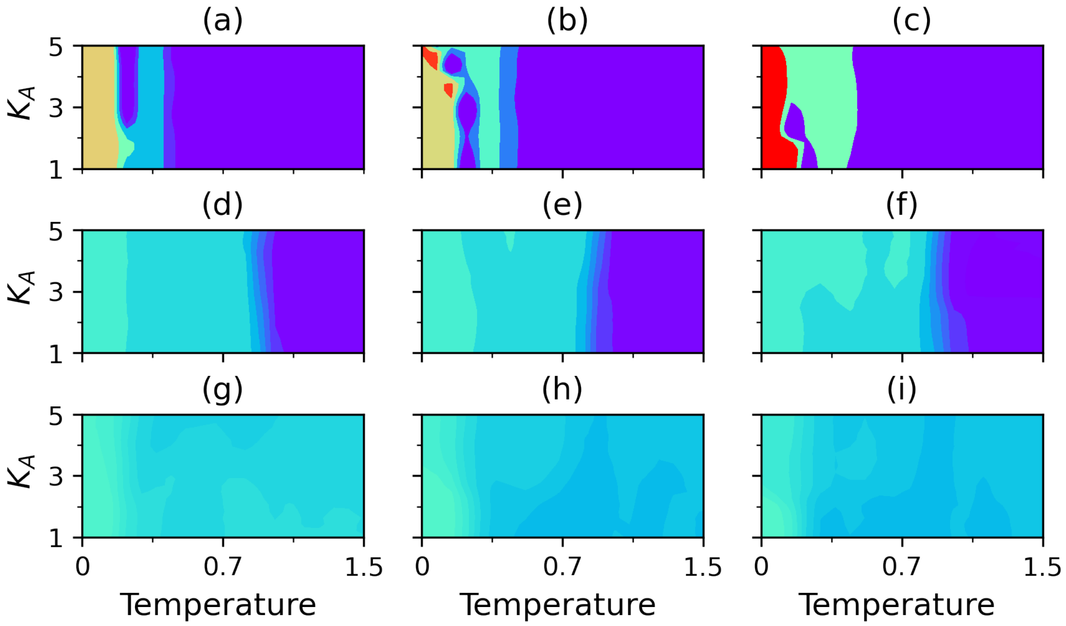


FIG. S3: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 64 monomers. Each plot shows the most probable polymer confirmation as a function of temperature and chain semiflexibility parameter $K_A$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $K_D \in \{1, 3, 5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

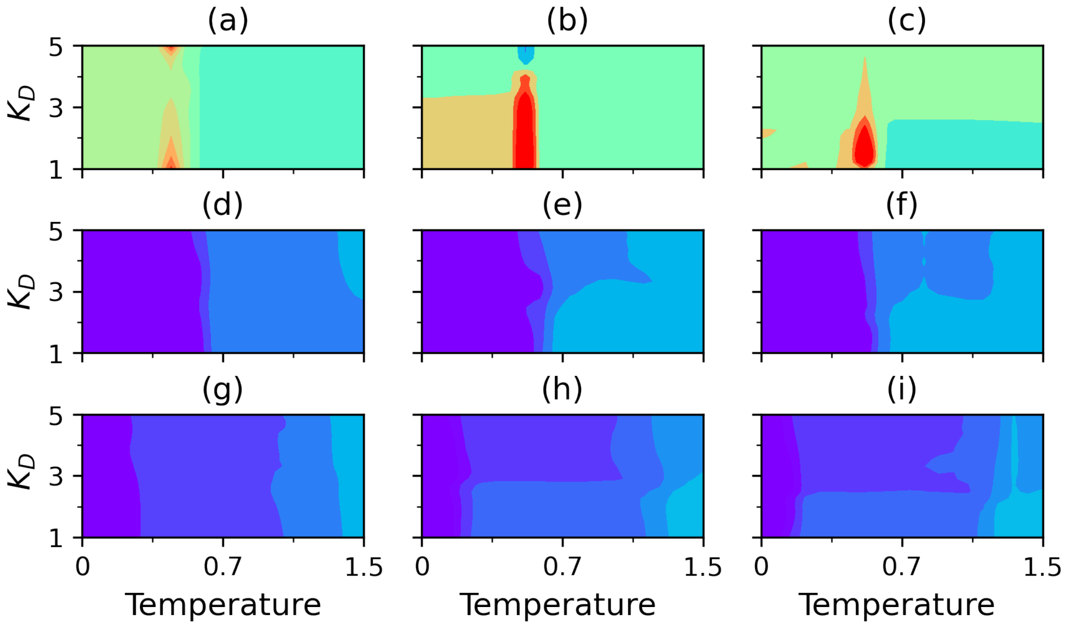


FIG. S4: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 22 monomers. Each plot shows the most probable polymer confirmation as a function of temperature and chain semiflexibility parameter $K_D$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $K_A \in \{1, 3, 5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

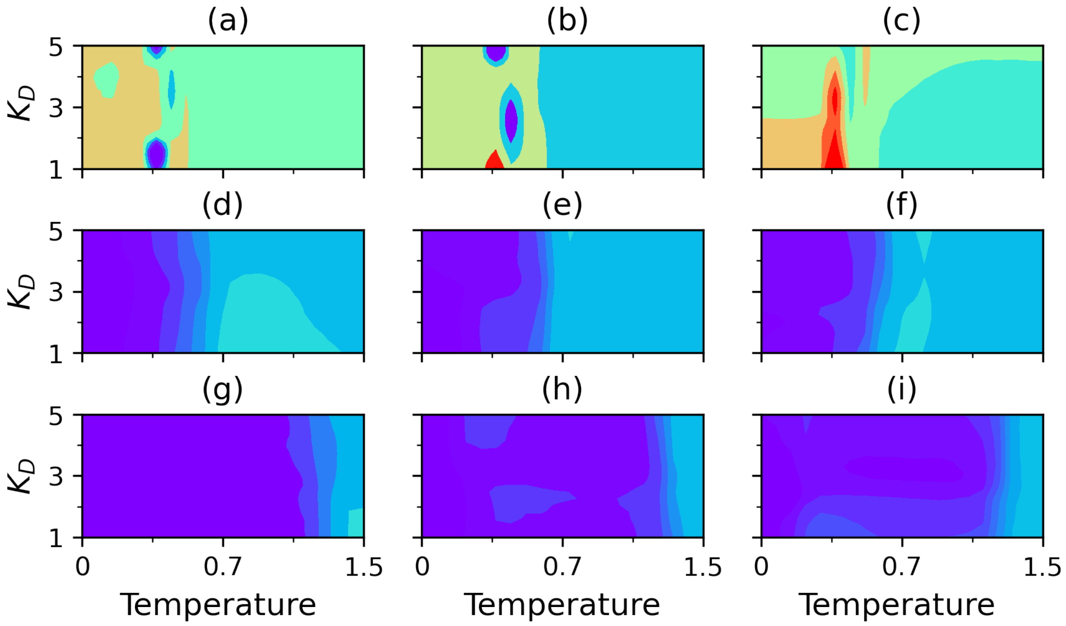


FIG. S5: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 44 monomers. Each plot shows the most probable polymer confirmation as a function of temperature and chain semiflexibility parameter $K_D$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $K_A \in \{1, 3, 5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

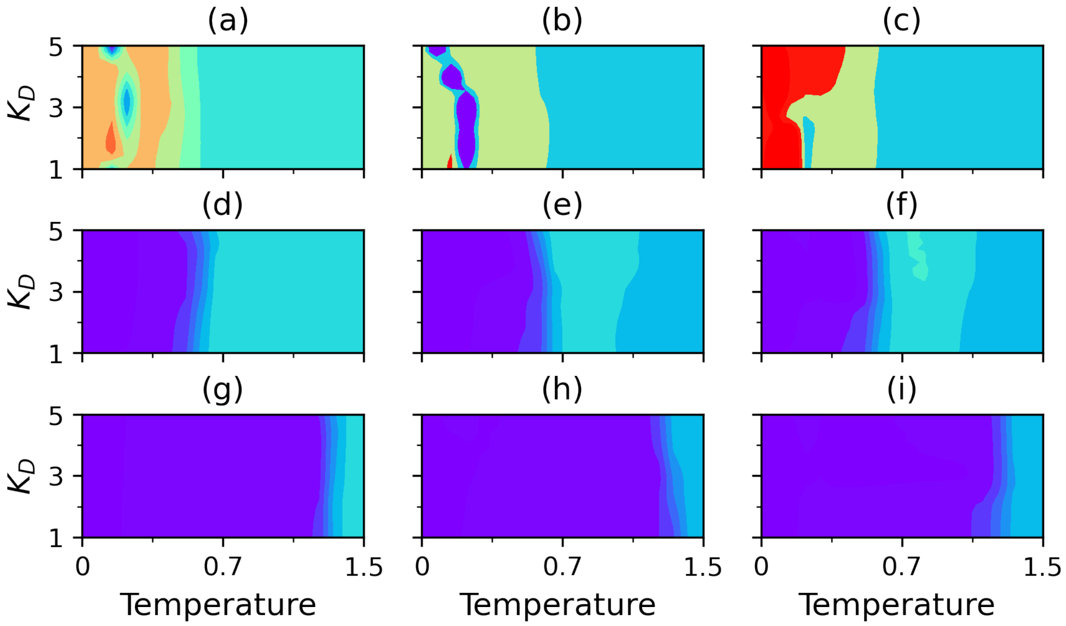


FIG. S6: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 64 monomers. Each plot shows the most probable polymer confirmation as a function of temperature and chain semiflexibility parameter $K_D$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $K_A \in \{1, 3, 5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

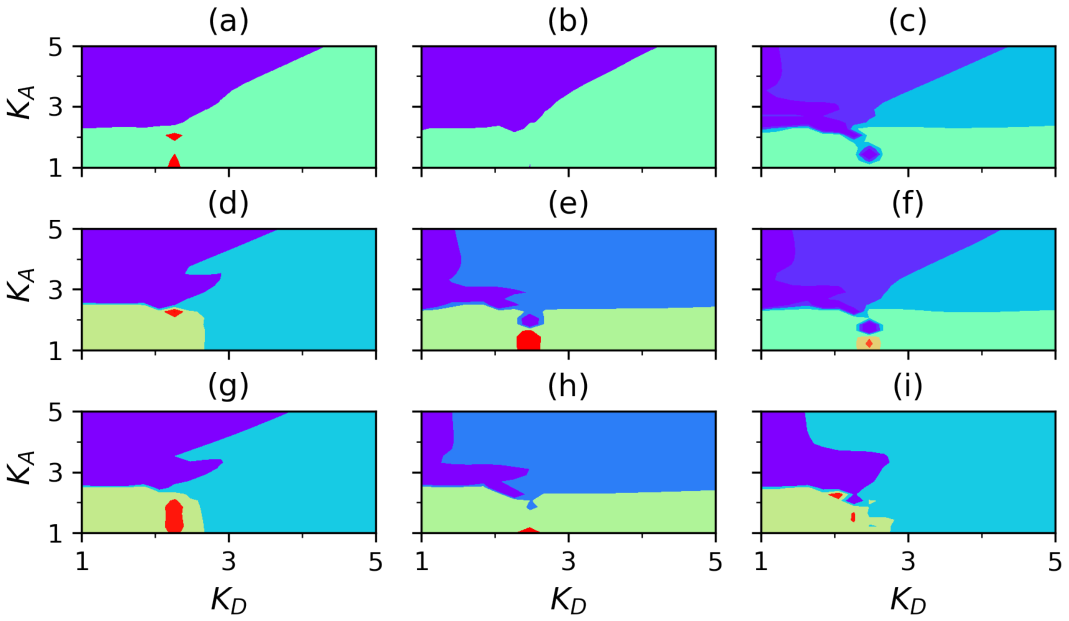


FIG. S7: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 22 monomers. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $T \in \{0.1, 0.6, 1.5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

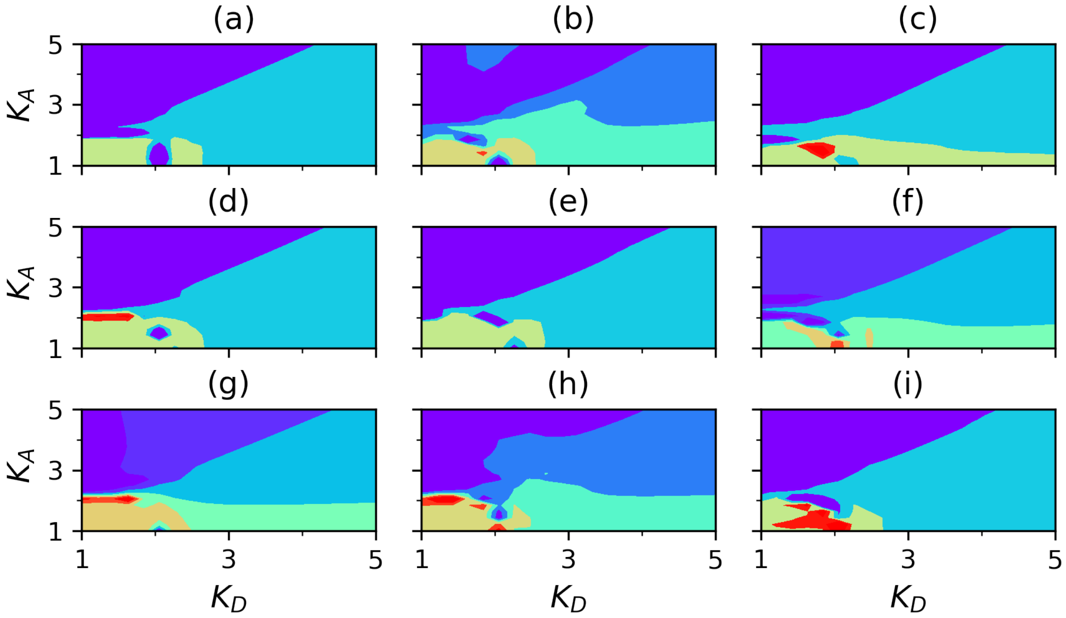


FIG. S8: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 44 monomers. Each plot shows the most probable polymer confirmation as a function of chain semiflexibility parameter $K_D$ vs. semiflexibility parameter $K_D$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $T \in \{0.1, 0.6, 1.5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.

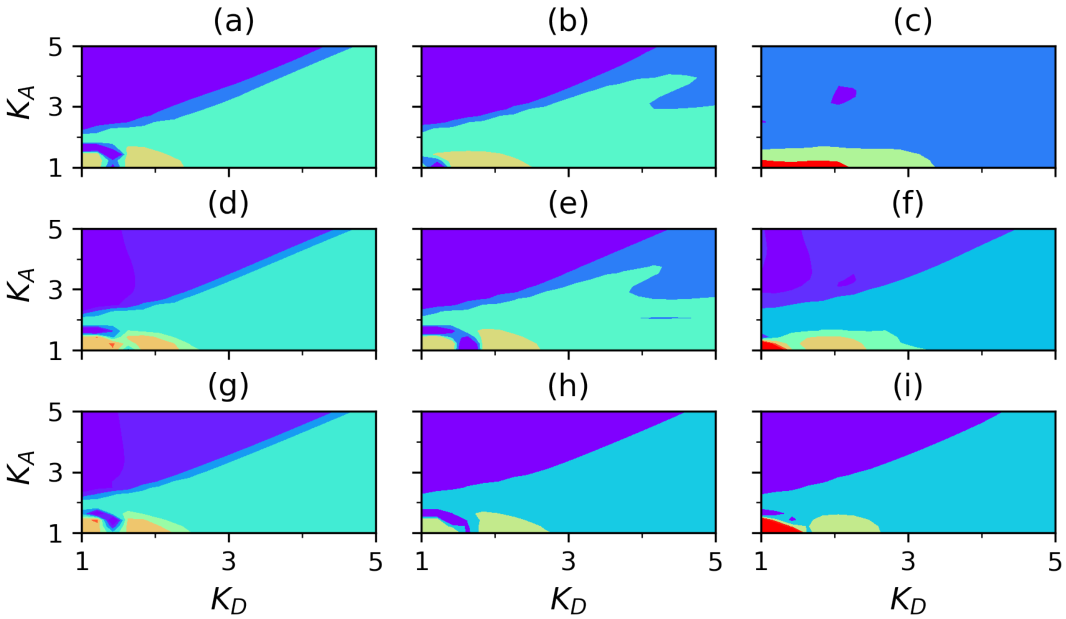


FIG. S9: Slices from the high dimensional phase diagram of the anisotropic polymer consisting of 64 monomers. Each plot shows the most probable polymer confirmation as a function of chain semiflexibility parameter $K_D$ vs. semiflexibility parameter $K_D$. Plots from left to right represent increasing $\tau \in \{500, 2500, 7500\}$ while plots from top to bottom show increasing $T \in \{0.1, 0.6, 1.5\}$. The range of the parameters of vector $\mathscr{D}$ have all been standardized between 0 and 1.