ACS OMEGA

http://pubs.acs.org/journal/acsodf

Mini-Review

# Machine Learning of Coarse-Grained Models for Organic Molecules and Polymers: Progress, Opportunities, and Challenges

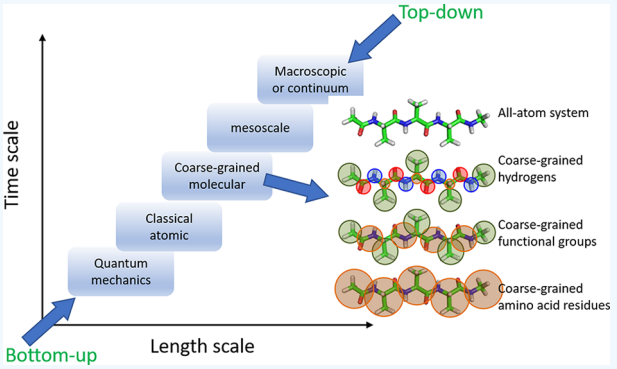
Huilin Ye,[§] Weikang Xian,[§] and Ying Li*

Cite This: *ACS Omega* 2021, 6, 1758−1772

Read Online

ACCESS | Metrics & More | Article Recommendations

**ABSTRACT:** Machine learning (ML) has emerged as one of the most powerful tools transforming all areas of science and engineering. The nature of molecular dynamics (MD) simulations, complex and time-consuming calculations, makes them particularly suitable for ML research. This review article focuses on recent advancements in developing efficient and accurate coarse-grained (CG) models using various ML methods, in terms of regulating the coarse-graining process, constructing adequate descriptors/features, generating representative training data sets, and optimization of the loss function. Two classes of the CG models are introduced: bottom-up and top-down CG methods. To illustrate these methods and demonstrate the open methodological questions, we survey several important principles in constructing CG models and how these are incorporated into ML methods and improved with specific learning techniques. Finally, we discuss some key aspects of developing machine-learned CG models with high accuracy and efficiency. Besides, we describe how these aspects are tackled in state-of-the-art methods and which remain to be addressed in the near future. We expect that these machine-learned CG models can address thermodynamic consistent, transferable, and representative issues in classical CG models.




## 1. INTRODUCTION

Over the past decades, molecular dynamics (MD) simulations have become one of the most important computational techniques to study the relationship between the properties of materials and the interactions of atoms, especially with the advances of computational resources, i.e., high-performance computing (HPC).[1] Although it has achieved great success, the progress driven by the demand to model more complex systems across multiple spatial and temporal scales is severely restrained by the limitations of computing aspects.[2] As shown in Figure 1a, the sweet spot for MD simulations is confined by the boundaries of memory limit (size of the system), communication limit (time across multiscale), and parallel limit (HPC). To model a larger system or a longer time, coarse-graining is required to overcome the current limitations and push the boundaries of MD simulations for a broad range of applications, such as self-assembly of organic molecules and crystallization of polymers.[3−5]

The central problem for coarse-graining is how to link simulations of detailed models with simulations of coarse-grained (CG) ones through the propagation of information. Note that the CG model enables efficient simulations covering different spatial and temporal scales, from the quantum mechanics with the highest level of accuracy to the macroscopic continuum model that depends on the theoretical constitutive law or experimental empirical relationship. According to the direction of information propagation, CG methods can be classified into two different categories: bottom-up[3] and top-down[6] approaches. For the bottom-up methods, the fundamental physical principles at the smaller scales are used to parametrize a model at a CG scale; while the behavior at larger scales is used to inform the interactions at smaller scales in top-down CG methods. When designing a CG model, the first step is to define a pseudo atom or CG site. These sites can be designed to represent combined groups of multiple atoms or functional groups. For example, Figure 1b shows the coarse-graining process of polyalanine. The pseudo atom sites can be either groups of hydrogens or functional groups. The second key aspect in developing the CG model is to define an effective energy function $U$ for the CG model, which determines the interactions between pseudo atoms.

In the top-down CG methods, the energy function $U$ is constructed and parametrized by either atomistic simulations or experiments to reproduce target properties, such as density,

Received: October 31, 2020
Accepted: January 4, 2021
Published: January 11, 2021







https://dx.doi.org/10.1021/acsomega.0c05321
*ACS Omega* 2021, 6, 1758−1772

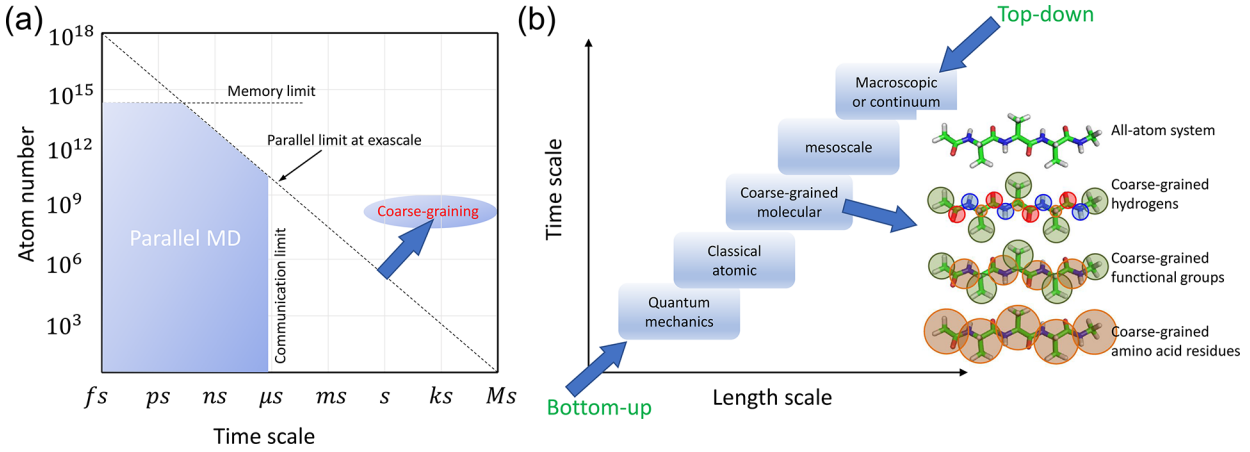



**Figure 1.** (a) Boundaries of all-atom molecular dynamics (MD) simulation using high performance computing facilities in terms of system size and time scale. (b) Schematic to show two types of coarse-grained methods: bottom-up and top-down in different time and length scales. Besides, the coarse-graining process of a polypeptide, polyalanine, is shown in the inset.

**Table 1. Classification of Different Machine-Learned CG Models**[a]

| category | name | machine learning technique | architecture hyperparameter selection | training data sets |
|---|---|---|---|---|
| top-down | BOP[9] | HOGA | two-stage optimization genetic algorithm | Experiment and atomistic simulation |
| | CG-LJ[10] | DNN | (48, 15) ground-truth error | MD simulation |
| | ANN-PSO[11] | ANN | (50, 50) property of force-match | MD simulation |
| bottom-up | DeepCG[12] | DNN | (120, 60, 30, 15) minimization of mean force | atomistic simulation |
| | CGnet[13] | ANN | (160, 160, 160, 160, 160) three-stage cross-validation | all-atom MD simulation |
| | Autoencoder[15] | auto-encoding | encoder and decoder Gubmel-softmax reparametrization | all-atom MD simulation |
| | Kernel-based[14] | kernel optimization | Hessian kernel mean force estimation | atomistic simulation |
| | Graph-based[16] | graph-theoretic principles | adjacency matrix structure of material | atomistic simulation |

[a] $(p, q)$ in the architecture column are the numbers of nodes in the layers of the neural network.

diffusivity, and partition energy. For example, Marrink et al.[6] developed the MARTINI model for biomolecular simulations, especially for the modeling of lipids and proteins. In the MARTINI model, the force field is parametrized in a systematic way to reproduce the free energies between polar and apolar phases of a large number of chemical compounds through increasing the number of possible interaction levels of the CG sites. On the contrary, in the bottom-up CG methods, the target properties are not used to optimize the potentials; instead, they are predicted by the CG potentials. Typical bottom-up methods use structure- and force-based approaches, depending on the target quantities. If the method aims to reproduce the structural distributions given by all-atom (AA) simulations, then it is structure-based, which is known as the iterative Boltzmann inversion (IBI) method, the inverse Monte Carlo (IMC) method, and the relative entropy method.[5] The force-matching method intends to match the force distribution of the CG model to that obtained from AA molecular simulations. Each method has its advantages and disadvantages: the IBI method can reproduce the correct structural distributions and conformations but misses the right thermodynamics and many-body free-energy landscape; while the force-matching method captures many-body effects and proper dynamics, it can misrepresent the structural distributions.

Although these CG models are straightforward and can model much larger systems with reasonable computation costs, they still suffer from some important issues in terms of thermodynamic representability, transferability, and consistency.[5] The CG model cannot adequately reproduce the properties of the AA system due to the degeneration of freedoms during the CG process. Some properties that are highly sensitive to small-scale phenomena will be lost in the simulations using CG models.[7] Thus, the CG process from AA coordinates to CG sites strongly depends on the understanding of the physical problem and chemical intuition, and a poor CG process will lose more important information. Besides, both CG methods take distributions from AA molecular simulations for a specific thermodynamic state. Therefore, the derived effective CG potentials have limited thermodynamic transferability to other conditions, such as temperature and pressure. It results in the limited applications of the CG potentials into a subset of thermodynamics states. Last but not the least, the thermodynamic consistency can be influenced by the CG process. The most common inconsistency is caused by the dilemma of balancing structural and dynamic properties. Taking the IBI method as an example, the forces that CG sites bear are not rigorously reproduced as the optimization objective is the preservation of the structural distributions, compared to the force-matching method. Consequently, the optimized CG model, based on the IBI method, usually shows faster dynamics. It requires a dynamic rescaling factor, despite faithfully preserving structural distributions and properties.[5]

Recently, machine learning (ML) of potentials is emerging as an alternative approach, in comparison with classical force fields with explicit functions. This method represents potential-energy surfaces by training large data sets from density-functional theory (DFT) calculations. Machine-learned AA potentials have excellent representability, accuracy, efficiency, and thermodynamic transferability, in comparison to DFT


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

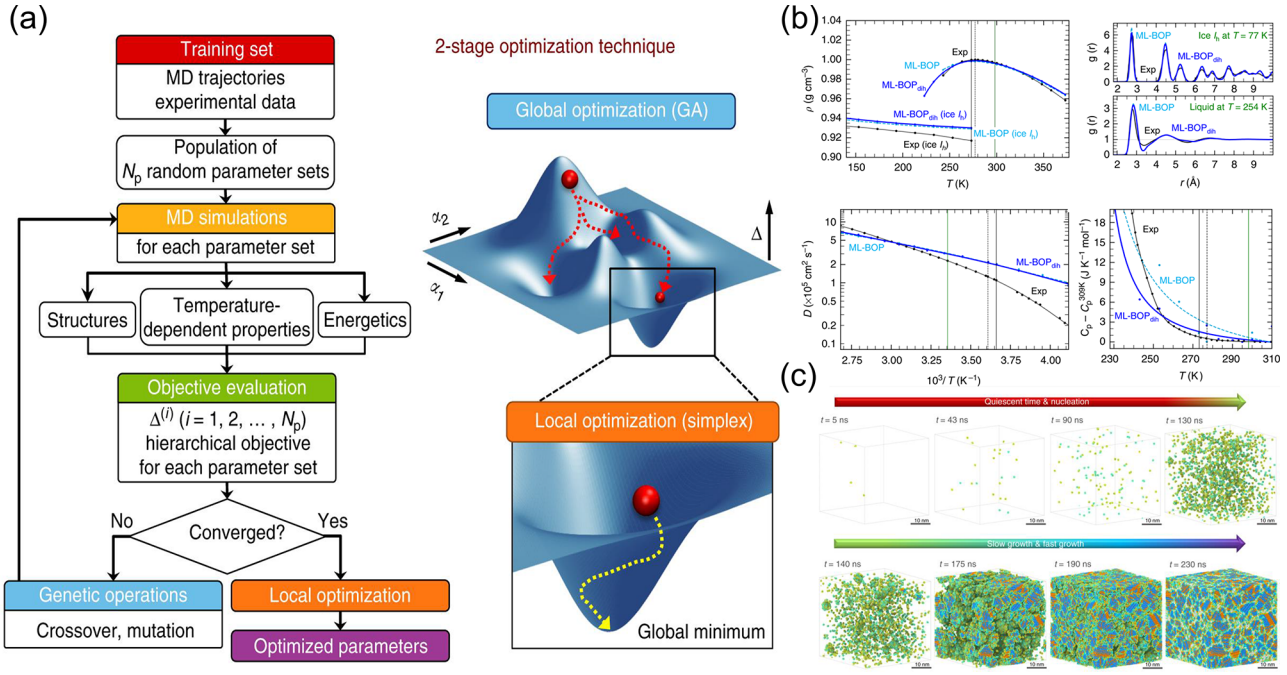



**Figure 2.** (a) Work flow for ML CG model of water molecules through hierarchical objective genetic algorithm. Two stages: global minimization and local minimization are shown. (b) Comparisons of the results of density anomaly, diffusion coefficients, radial distribution functions and heat capacity between the experiments and predictions by ML CG model. (c) Application of ML model to predict the nucleation and crystallization process of water (Reprinted with permission from the work of Chan et al.[9] Copyright 2019 Springer Nature Limited).

simulations. Similarly, machine-learned CG potentials are proposed to represent the free-energy landscape of AA molecular models with high efficiency and accuracy. Thus, ML of CG potentials becomes a new way to bridge the gap between accurate but computationally expensive *ab initio* methods and approximate but computationally cheap CG method.[8−19] Particularly, machine-learned CG potentials can predict the physical properties of complex molecular systems with *ab initio* accuracy. The general idea of this approach is to use statistical learning techniques, such as artificial neural network (ANN),[13] deep neural network (DNN),[12] and convolutional neural network (CNN),[18] to name a few, to formulate complex potentials with many model parameters, which are undetermined and optimized using *ab initio* or AA molecular simulation results as references. Similarly, based on the classes of CG methods, the machine-learned CG methods have two categories: bottom-up machine-learned CG method[9−11] and top-down machine-learned CG method,[12−16] as listed in Table 1.

For the top-down machine-learned CG method, an empirical expression of the interactions between CG particles is given *in prior*, such as Tersoff−Brenner potential for CG water[9] and Lennard-Jones potential for simple molecular liquids,[10] to perform MD simulations. Usually, the potential parameters in these expressions are undetermined and optimized by statistical learning techniques to match the structural properties (e.g., radial distribution functions, angular distribution functions) and dynamics properties (e.g., diffusion coefficient) with the corresponding experimental/computational results. On the contrary, in the bottom-up machine-learned CG method, an unknown CG potential $U$ should be introduced and $U$ is related to the representations (features associated with the coordinates of CG particles) through high-dimensional neural networks.[12] The neural networks can be trained and optimized by using the data set from AA molecular simulations. For instance, the mean force is used to regularize the optimization of the loss function in the learning process. In this study, we will review some of the typical machine-learned CG methods in these two categories. And we will focus more on the ML framework, the validation of the trained model, and its applications. We will also discuss the open problems and challenges facing the ML of CG models and possible solutions. We expect machine-learned CG models can play an important role in understanding the physical and mechanical properties of complex molecular systems and hope that this work inspires future studies in this field.

## 2. TOP-DOWN MACHINE-LEARNED CG METHODS

**2.1. Machine-Learned Bond Order Potential for CG Water.** Accurate and efficient molecular models for water molecules are highly desirable to understand their phase transformation behaviors at different temperatures or pressures. However, it remains challenging to develop robust and efficient molecular models for water molecules. Recently, Chan and co-workers[9] have introduced an ML workflow to train the CG models that can accurately describe the behaviors of ice, supercooled, and normal liquid water at the mesoscopic scale. Two bond-order CG models are utilized: bond-order potential (BOP) and BOP with on-the-fly dihedrals ($BOP_{dih}$), which are both based on the Tersoff−Brenner potential with 7 and 11 parameters undetermined and to be optimized, respectively.

In conventional MD simulations, these parameters in the potential functions are given *a priori*. It limits the application of the potential, and modeling systems at different thermodynamic states requires the recalibration of the potential parameters. In this study, the CG potential parameters are optimized by a multilevel evolutionary strategy (hierarchical objective genetic algorithm−HOGA) as shown in Figure 2a. A two-stage optimization technique is introduced to find the


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

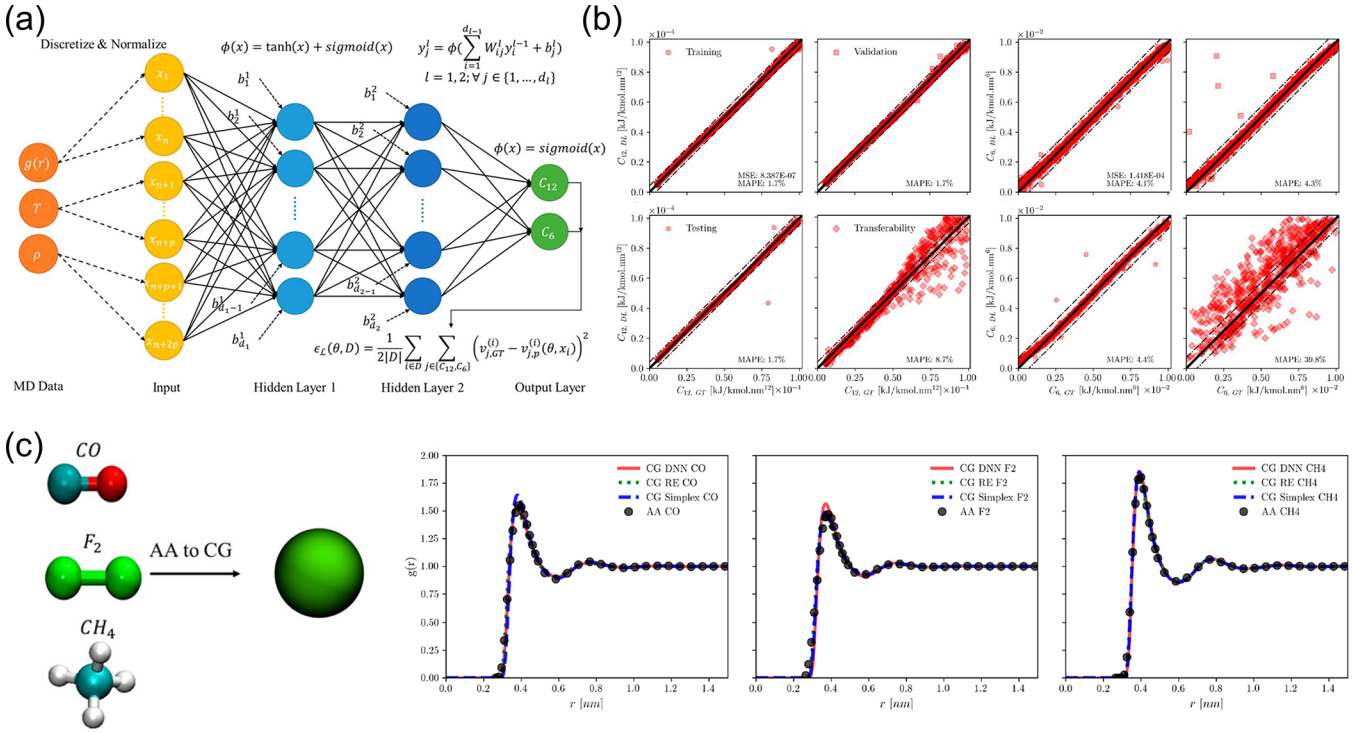



**Figure 3.** (a) Schematic of a deep neural network (DNN). (b) Comparison of the pair potential parameters determined from the DNN with the ground-truth values for training, validation, testing, and transferability data sets. (c) Comparison of RDFs obtained with DNN, relative entropy, and simplex CG models and the all-atom (AA) model. Three types of molecules are studied: CO, $F_2$, and $CH_4$ (Reproduced from the work of Moradzadeh et al.[10] Copyright 2019 American Chemical Society).

global minimum in the objective value landscape. During the global optimization, the genetic algorithm[20] is used. It begins with the random sampling of the parameter sets $N_p$. The objective value for each of these parameter sets is evaluated, and the convergence is checked until the convergence criteria are met. If not, a new list of $N_p$ is generated using genetic operations. Typically, this stage can return a list of close-to-optimal parameter sets. And using these parameter sets, the second stage local optimization technique starts to search the final parameter sets with the Nelder–Mead simplex algorithm.

The extensive training data set of energies and structural properties for ice and liquid water are taken from both the atomistic model (TIP4/2005) through MD simulations and experimental data. The data set is chosen with the requirement to ensure the adequate representation of the diverse configurational space of ice and liquid water. The trained CG model is first validated by comparing the structural and dynamic properties with experimental data, such as radial distribution function (RDF), angular distribution function (ADF), and diffusion coefficient. From Figure 2b, it is found that the ML-BOP models can successfully capture the best-known thermodynamic anomaly and the existence of a density maximum at 277 K. Also, it can correctly describe the freezing/melting transition at 273 ± 1 K, densities of ice (140–273 K), and water (243–273 K) within 1.4% of experiments. For the transport properties of water, the room temperature diffusivity is calibrated as $\sim 3 \times 10^{-5}$ cm$^2$ s$^{-1}$, which is close to that in experiments ($2.3 \times 10^{-5}$ cm$^2$ s$^{-1}$). Regarding the structural property, the O–O RDFs for ice at 77 K and water at 254 K are compared with the experiments. As shown in Figure 2b, the location and intensities of the peaks of these RDF are in good agreement with the corresponding experimental results. Finally, the heat capacities $C_p$ for liquid water are investigated. It can reproduce the thermodynamic anomaly indicated by the sharp increase in $C_p$ of supercooled water (cf. Figure 2b). After the validation, a representative test case is performed on multimillion water molecules to study nucleation of supercooled water, leading up to the formation and growth of grains as shown in Figure 2c. The water is slowly cooled from 275 to 210 K over 130.4 ns. When the first nucleation forms, the temperature is held at 210 K for another 100 ns, and the first nucleation event is followed by a slow transformation (150 ns), an accelerated transformation of a large number of nuclei (200 ns), and completion of grain growth to form polycrystalline ice.

This workflow demonstrates the high accuracy of the machine-learned CG model in capturing the structural and dynamic properties of water, in comparison to the experimental results. Besides, this workflow shows almost the same computational cost as the other CG method—the monatomic water (mW) model—but with higher accuracy. Nevertheless, this workflow suffers from some limitations. First, it is partially temperature-transferable. It can only reproduce the thermodynamics within the regime which is covered in the training data set. Second, it cannot fully reproduce all the physical properties. For example, to improve the predictions on diffusion coefficient and temperature of maximum density, the prediction accuracy of the melting point is sacrificed.

**2.2. Transfer-Learning-Based CG Method for Simple Molecular Fluids.** In MD simulations, once the interatomic potential is specified, many properties like RDF and ADF are determined. However, the inverse problem—parametrization of the potential for a specific property is not straightforward.


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758–1772

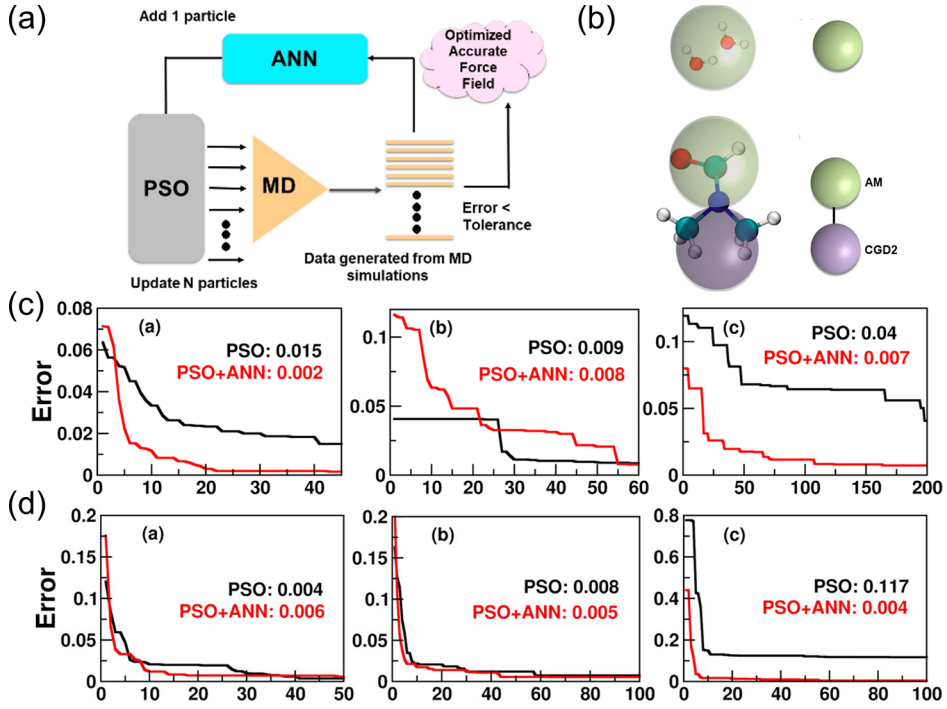



**Figure 4.** Artificial neuron network (ANN) assisted particle swarm optimization (PSO). (a) Flowchart of the ANN-PSO. In every iteration, the ANN trained by data collection of all the previous iterations provides a new particle (a set of force field parameters). (b) Molecules $D_2O$ and DMF coarse-grained to nonpolar beads. (c) Optimization of the $D_2O$. Error histories of systems with 40, 8, and 4 particles. (d) Optimization of the DMF. Error histories of systems with 40, 8, and 4 particles (Reproduced from the work of Bejagam et al.[11] Copyright 2018 American Chemical Society).

Moradzadeh et al.[10] adopted DNN to study this inverse problem, in terms of the relation between the RDF and the Lennard-Jones (LJ) potential parameters. Figure 3a shows the schematic of the DNN. The training data is taken from MD simulations with the potential parameters and thermodynamic states (density $\rho$ and temperature $T$) sampled over a wide range. Here, the LJ potential has the form

$$u(r) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \tag{1}$$

where $C_{12}$ and $C_6$ are potential parameters. After performing the MD simulations, the RDFs $g(r)$ under specific thermodynamic states ($\rho$, $T$) are collected. And the complex relationship between potential parameters ($C_{12}$ and $C_6$) and RDFs can be expressed as $(C_{12}, C_6) = \mathbf{f}(g(r); \rho, T)$, where $\mathbf{f}$ is a vector function. In this work, a feed-forward neural network (FFNN) is adopted to represent this function based on the universal approximation theorem as $x_i = (\mathbf{g}_i(r), \rho_i^1, \rho_i^2, ..., \rho_i^p, T_i^1, T_i^2, ..., T_i^p,)_m$, where $x_i$ is the input vector composed of the concatenation of system $i$ RDF (size of $n$) and thermodynamic states (each with a size of $p$) in the data set with a total size of $m$. The node in DNN applies a linear transformation after receiving the input signal and is followed by a nonlinear activation function, resulting in an output signal. Finally, the DNN can be mathematically expressed as

$$(C_{12}^{\text{fnn}}, C_6^{\text{fnn}}) = \phi_0(\mathbf{W}_0\phi_n(...\phi_2(\mathbf{W}_2\phi_1(\mathbf{W}_1\mathbf{x}_i + \mathbf{b}_1) + \mathbf{b}_2)) + \mathbf{b}_0) \tag{2}$$

where $\phi_k$ is the nonlinear activation function of layer $k$, $\mathbf{W}_k$ and $\mathbf{b}_k$ are weights and biases. The FFNN is trained based on the loss function: mean-squared error (MSE) between the ground-truth and the predicted parameters

$$\epsilon_L(\theta, D) = \frac{1}{2|D|}\sum_{i\in D}\sum_{j\in C_{12},C_6}(\nu_{j,\text{GT}}^{(i)} - \nu_{j,p}^{(i)}(\theta, x_i))^2 \tag{3}$$

where $\theta$ represents the free parameters like the weights and biases. $\nu_{j,\text{GT}}^{(i)}$ and $\nu_{j,p}^{(i)}(\theta, x_i)$ are the ground-truth and predicted LJ interaction parameters.

The performance of DNN is examined by investigating two cases. The first one is the generalizability and transferability of the interatomic potential parameters. The generalizability refers to the use of DNN to estimate the case, which is not part of the training data set, but falls within the range of the training data set; the transferability refers to the use of DNN to estimate the case, which is outside the range of the training data set. One-to-one comparison between the ground-truth (denoted by subscript GT) and predicted (denoted by subscript DL) potential parameters are shown in Figure 3b. The left four and right four panels are for $C_{12}$ and $C_6$, respectively. In each panel, the first one is for the training process; the second one shows the validation; the third one gives the generalizability, and the last one demonstrates the transferability. The accuracy is quantified by introducing the mean absolute percentage error (MAPE), defined as $\epsilon_{\text{MAPE},j} = 100 \times \frac{\sum_{i\in D}|\nu_{j,\text{DL}}^{(i)} - \nu_{j,\text{GT}}^{(i)}|}{\sum_{i\in D}|\nu_{j,\text{GT}}^{(i)}|}$. The solid black line represents that the predicted parameters are exactly consistent with the ground-truth parameters. The dashed black lines are the boundaries in which the ~99% of prediction points are located, in comparison to the ground-truth of the training data set. The low MAPE in Figure 3b demonstrates the DNN has good generalizability and transferability.

The second one is transfer learning, which is studied by developing single-bead CG models for simple molecular liquids such as carbon monoxide, fluorine, and methane. The


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

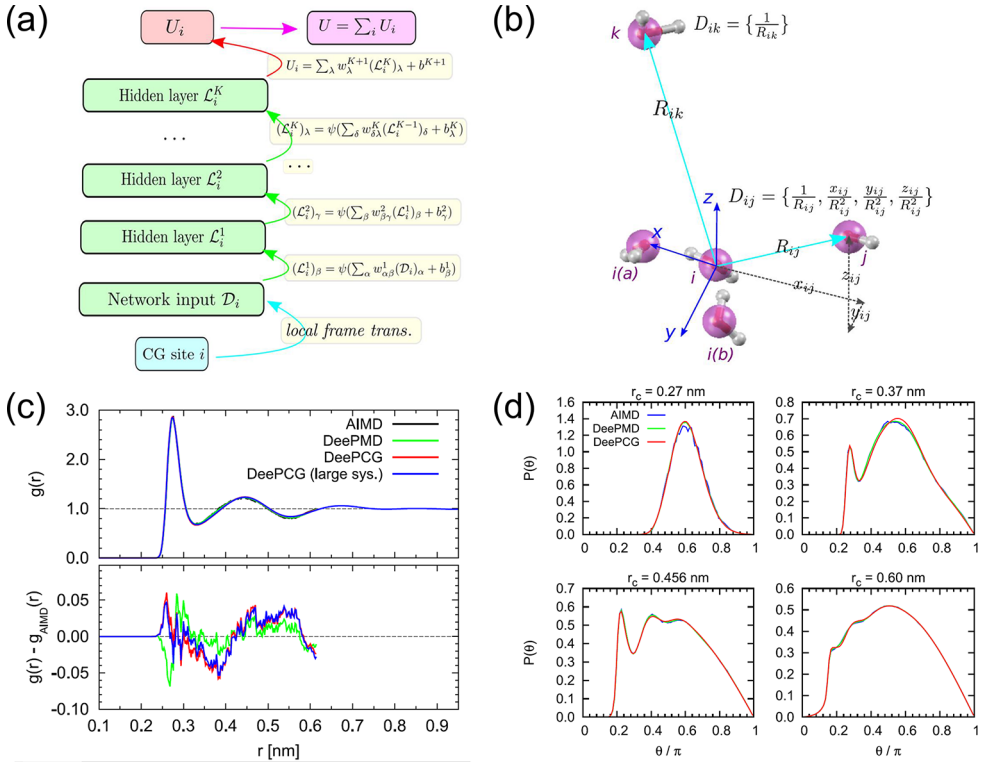



**Figure 5.** (a) Schematic of the deep neural network (DNN). (b) Example for CG of water molecules and the local transformation of coordinates of CG sites. $\{\mathbf{D}_{ik}\}$ and $\{\mathbf{D}_{ij}\}$ are radial and angular descriptors, respectively. (c) (upper) comparison of the O−O RDFs for liquid water between AIMD and DeePMD and DeePCG simulation; (lower) deviations of DeePMD and of two DeePCG models relative to the AIMD result. (d) O−O−O ADFs of liquid water from AIMD, DeePMD, and DeePCG simulations with four different cutoff radii (Reprinted with permission from the work of Zhang et al.[12] Copyright 2018 AIP Publishing LLC).

corresponding RDFs predicted from DNN, simplex, relative entropy method, and AA molecular simulations are shown in Figure 3c. It indicates that DNN prediction has a high accuracy from these comparisons. However, this high accuracy is only limited to simple molecular liquids. For complex molecules that cannot be described by the simple LJ potential functions, the CG process to a single LJ particle will lose plenty of information, leading to the wrong predictions of both structural and dynamic properties.

**2.3. ANN-Assisted Particle Swarm Optimization for Transferable CG Models.** Particle swarm optimization (PSO), inspired by Mother Nature and first applied to animate social behaviors, is an iterative global optimization technique powered by population migration. Although it is capable of nonlinear optimization, the convergence becomes slow when applied to force field development because of the increasing amount of expensive MD simulations. In addition, due to its population-based nature, its optimization path relies on only the local bests and the global bests, instead of all the MD results. Therefore, a cost-effective way to harness all the costly simulation data is desirable. Recently, Bejagam et al.[11] developed an artificial neural network (ANN) assisted-PSO framework to optimize the potential parameters for CG models.

In a traditional PSO optimization, a particle represents a state in the multidimensional optimization space. In the case of CG potential development, the state represents the potential parameters. In each iteration of the PSO, MD simulations are first performed with particle-specified potential parameters. By comparing each of the MD results and target properties, a fitness value, which quantifies the particle-specific deviation from the target properties, is assigned to the particle as a personal score. For each particle, the local best will be updated if the current fitness value is higher than all the previous values. The best of all the local bests is assigned as the global optimized point for the current iteration and used to guide the next iteration. The process of optimization is performed until the global best particle (the best potential parameters) being capable of producing the target properties with a satisfactory small, usually 2−5%, error.

On top of the classic PSO framework, an ANN is used to accelerate the convergence of the PSO. As shown in Figure 4a, in each iteration of the PSO, after the MD runs, all the MD data are used to train a dynamic ANN. The ANN then makes a prediction of an extra particle (a set of potential parameters). This extra particle is fed back to the group of particles in the PSO. In the next iteration, the new group of particles is used to advance the PSO. Thus, the combination of PSO and ANN algorithms could dramatically accelerate the searching process for the best solution for CG potential parameters.

Molecules of $D_2O$ and DMF, as depicted in Figure 4b, are used to demonstrate the competence of the proposed framework. The molecules are coarse-grained as one and two nonpolarizable beads, respectively. The single-bead CG model for the $D_2O$ needs two parameters, i.e., the $\epsilon$ and $\sigma$, in 12−6 LJ potential, to descried the interaction between beads. The model for DMF needs five parameters: $k_b$ the strength describing the bonding between the AM beads and the CGD2 beads, $\epsilon_{AM}$ and $\sigma_{AM}$ for the AM beads, and $\epsilon_{CGD2}$ and $\sigma_{CGD2}$ for the CGD2 beads respectively. The results of the optimization for the $D_2O$ and the DMF are shown in Figure 4c and d, respectively. The error of each iteration is calculated as


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

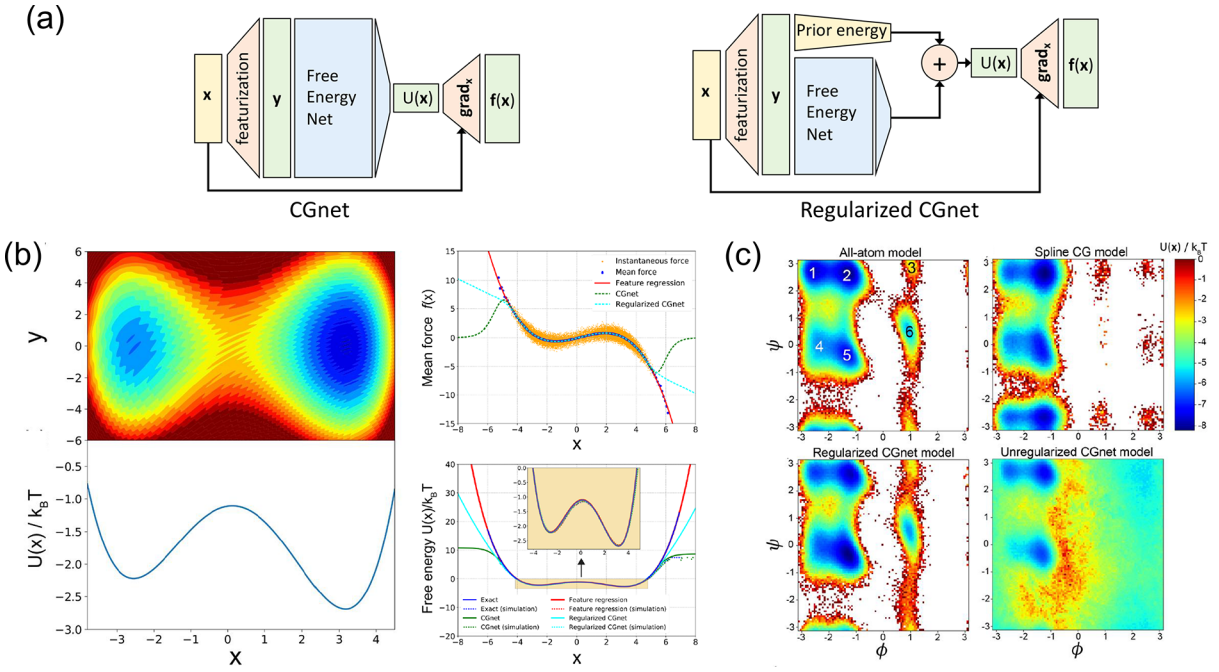



**Figure 6.** (a) ML schemes of CGnet and regularized CGnet. (b) Comparisons of force and free energy between predictions from CGnet, feature regression, and the exact results for the two-dimensional toy model. (c) Free energy profiles of alanine dipeptide using all-atom and machine-learned CG models (Reproduced from the work of Wang et al.[13] Copyright 2019 American Chemical Society).

the root-mean-square error between the optimization and the experimental target values of the densities and diffusion coefficients. The particle numbers used in the optimization are 40, 8, and 4 from the left to the right, respectively. In all the cases, the ANN-PSO method shows better prediction and fast convergence with smaller errors than the PSO method.

In this study, the ANN is used to exploit all the MD results that the PSO produces to accelerate the convergence of the PSO, reducing the demand for expensive MD simulations. However, because one still needs to predefine the CG potential forms (12−6 LJ form in this study). Generally speaking, this framework can only preserve limited structural and dynamic properties for the CG models.

## 3. BOTTOM-UP MACHINE-LEARNED CG APPROACHES

### 3.1. DeepCG: Constructing a CG Model via DNN.

Defining an accurate free energy function in the space of CG variables is always the key to developing a CG model. But, it is the most difficult part, which requires substantial physical and chemical intuitions. ML methods can address this problem in a more accurate and automated way. Nevertheless, most of the ML approaches so far focus on the representation of potential energy surface, such as the deep potential method.[21] It allows us to perform MD simulations with comparable accuracy as the *ab initio* molecular dynamics (AIMD) but only at the cost of classical empirical force fields. To find a good ML approach to represent the free energy surface, Zhang et al.[12] developed the deep coarse-grained potential (DeePCG) scheme. Note that the ML of AA and CG models tends to represent potential-energy and free-energy surfaces, respectively, by high-dimensional neural networks.

In DeePCG, a neural network representation $U^w(\boldsymbol{\xi})$ is adopted for the CG potential $U(\boldsymbol{\xi})$ (see Figure 5a). Here, $\boldsymbol{\xi}$ variables are the coordinates of the CG particles. The CG potential is assumed to be the sum of the local contributions of the CG particles, i.e., $U^w(\boldsymbol{\xi}) = \sum_i U_i^w(\boldsymbol{\xi})$, $U_i^w(\boldsymbol{\xi})$ is the potential contribution of the CG particle $i$. It is constructed in two steps: (1) The first is local frame transformation according to the descriptor $\{\mathbf{D}_{ij}\}$, which includes radial distance $\frac{1}{R_{ik}}$ and angular information $\left\{\frac{1}{R_{ij}}, \frac{x_{ij}}{R_{ij}^2}, \frac{y_{ij}}{R_{ij}^2}, \frac{z_{ij}}{R_{ij}^2}\right\}$ (Figure 5b). (2) The descriptor $\{\mathbf{D}_{ij}\}$ will be given as input to a fully connected FFNN (total of four layers with two hidden layers) to compute the potential contribution of the CG particle $i$ (see Figure 5a). Because the CG potential in the DeePCG is free energy that is not directly available, the optimization of CG potential is required. Here, the force-matching scheme is employed through fitting accurate mean forces from AIMD simulations. Therefore, the loss function in DeePCG is defined as

$$L(\mathbf{w}) = \frac{1}{dDM} \sum_{n=1}^{D} \sum_{i=1}^{dM} |F_i(\xi_n) + \partial_i U^w(\xi_n)|^2 \tag{4}$$

where $D$ is the number of configurations for CG variables in the data set and $F_i(\xi_n)$ is the mean force estimated from AIMD simulations. The optimization of the loss function is fulfilled by the stochastic gradient descent (SGD) method, which is a highly noncanvex function corresponding to a rugged landscape in the large parameter space.

To demonstrate the capability of the DeePCG, the CG model of liquid water is adopted. The training data set is obtained from the AIMD simulations. The AIMD data set consists of a total of 40 000 snapshots, during a 20 ps-long trajectory in the *NVT* ensemble with the system size $N$ = 192 atoms (64 $H_2O$ molecules) at $T$ = 300 K. After the training process, the NVT simulation with the trained DeePCG model is performed on the CG waters. The results of O−O RDF and O−O−O ADFs are compared with those from AIMD and


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

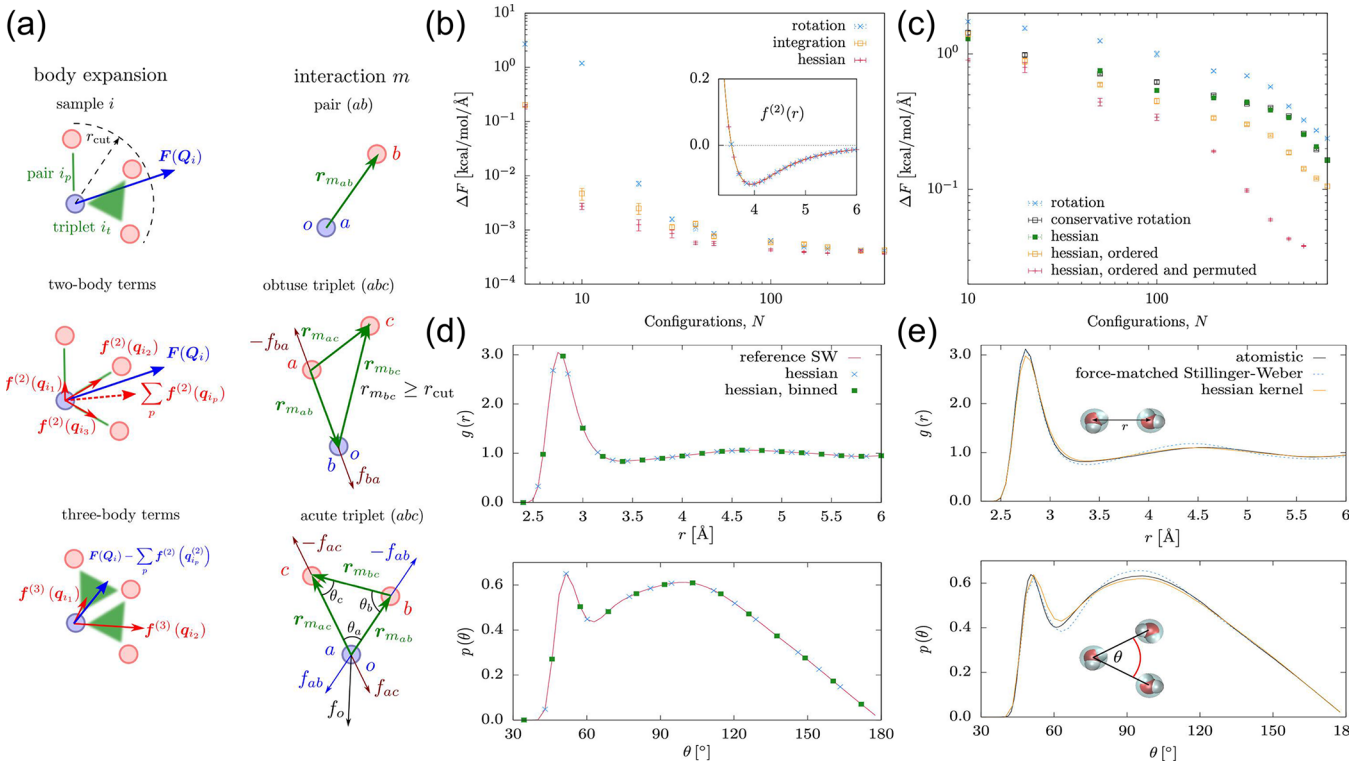



**Figure 7.** (a) (left) Description of the body expansion on pairs and triplets. Two- and three-body forces are shown. (right) Representations used for two- and three-body interactions. (b) Learning curves for two-body LJ fluid with different kernel matrices. (c) Learning curves for three-body SW fluids with different kernel matrices. (d) Comparison of MD simulation of tabulated three-body kernel predictions with the reference SW simulation, including RDF and ADF results. (e) Comparisons of RDF and ADF between atomistic simulations, the CG model using the force-matching scheme, and the SW model (Reproduced from the work of Scherer et al.[14] Copyright 2020 American Chemical Society).

DeePMD (see Figure 5c). It is found the DeePCG can reproduce the same RDF as those from AIMD and DeepMD. Also, the ADF results are examined for the CG system with different cutoff $R_c$ values as shown in Figure 5d. In DeepCG, the CG site is the oxygen, and the cutoff determines the maximum distance where the surrounding CG particles can interact with each other. It is found that, with the increase of the cutoff distance $R_c$, the ADF from DeePCG will be more consistent with those from AIMD and DeePMD. It means that a short cutoff distance may lose information contained within the interactions among the CG particles.

From the modeling of liquid water, DeePCG demonstrates the same accuracy as AIMD. And it performs faster than the corresponding AIMD since DeePCG only considers the local contributions of the potential functions, similar to classical atomistic MD simulations (see Figure 5b). Nevertheless, in the DeePCG simulations, the force discontinuity is found due to the sharp cutoff and limitation of the descriptors for angular information (see ADF results in Figure 5d). Moreover, the thermodynamic transferability of the developed CG model remains to be further examined with DeePCG.

**3.2. CGnet: ML CG Force Fields.** Another force-matching based ML approach for CG potential is named CGnet, as shown in Figure 6a. The difference is that in the CGnet, the CG potential $U(x)$ is represented by ANN. Similarly, the ANN includes a transformation $\mathbf{y} = g(\mathbf{x})$ from CG particle's coordinates $\mathbf{x}$ to a set of features $\mathbf{y}$ that contains the invariance of the free energy. The invariance is conserved by using the features: distances between all pairs of CG particles and the angles between three consecutive CG particles like those shown in Figure 5b. The features $\mathbf{y}$ are then given as inputs of the ANN. The loss function in ANN is defined:

$$L(\theta, \mathbf{R}) = \frac{1}{3Mn}\sum_{i=1}^{M}\left|\xi(\mathbf{F}(\mathbf{r}_i)) + \Delta U(\xi(\mathbf{r}_i); \theta)\right|^2 \quad (5)$$

where $\theta$ are parameters used in ANN, $\xi(\mathbf{R})$ are matrices of CG particle's coordinates, and $\xi(\mathbf{F}(\mathbf{R}))$ are instantaneous force components projected on the CG coordinates, with $\mathbf{F}(\mathbf{R})$ defined as the instantaneous atomistic forces. This loss function reflects the potential mean force (PMF) error between the mean force $\xi(\mathbf{F}(\mathbf{R}))$ and the CG force predicted by the gradient of CG potential—$\Delta U(\xi(\mathbf{r}_i); \theta)$. To minimize this error, another gradient layer is added into the ANN to compute the derivatives with respect to the input coordinates/features. With this ANN at hand, the trained CG potential form can be used to perform MD simulations that will produce new CG coordinates. When part of the new coordinates are outside the training data set, it is possible that the ANN can generate unphysical predictions. To avoid the unphysical predictions, a regularized CGnet is introduced by utilizing "regularization" methods (see Figure 6a). In the regularized CGnet, a baseline (prior) energy $U_0(\mathbf{x})$ is added into the energy function as $U(\mathbf{x}; \theta) = U_0(\mathbf{x}) + U_{net}(\mathbf{x}; \theta)$, where $U_{net}$ is the neural network free energy as used in CGnet. The role of $U_0(\mathbf{x})$ is to enforce $U \to \infty$ for unphysical states, in which the new CG coordinates are outside the training data set.

A two-dimensional toy model is used as an illustration to show the predicted potential energies by using different CG models. The two-dimensional potential energy has the analytical form


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

$$\frac{V(x,y)}{k_B T} = \frac{1}{50}(x-4)(x-2)(x+2)(x+3) + \frac{1}{20}y^2 + \frac{1}{25}\sin(3(x+5)(y-6))$$

, shown in Figure 6b. The CG mapping is defined by the projection of a two-dimensional model onto the $x$-axis (see Figure 6b). A long simulation trajectory of the two-dimensional model is obtained, which is used as the training data set. The CG potential and mean force are compared between those from feature regression, i.e., least-squares regression, and those from CGnet and regularized CGnet. It is found that within the training data set, both feature regression and CGnets (both CGnet and regularized CGnet) can accurately capture the mean force and free energy. However, outside the training data set, the predictions from CGnet severely deviates from the exact results, but the regularized CGnet, to some extent, can still nearly capture the exact results due to the introduction of the prior energy (here is a harmonic energy form). To further demonstrate the application of CGnet, the coarse-graining of alanine dipeptide in an explicit solvent is studied. To make a comparison, another CG model named the "spline model" is also studied.[2] The free energy profiles from these three CG models are displayed with the result from the AA molecular model (see Figure 6c). It can be found that only the regularized CGnet model can correctly reproduce the position of all the main free-energy minima. On the contrary, the spline model cannot capture the energy minimal corresponding to the positive value of the dihedral angle $\phi$, and the CGnet can only reproduce part of the free energy minima.

The above results demonstrate that the CGnets (CGnet and regularized CGnet) can be used to reproduce effective free energies for CG models, which can capture the equilibrium distribution of a specific atomistic model. However, the CGnet here is not transferable to the study of different systems, since it is designed *ad hoc* for a specific molecule. Besides, the CGnet can only reproduce the structural/configurational properties of the system with the dynamic properties remained to be explored using alternative approaches.

**3.3. Kernel-Based ML of the CG Model for Efficient MD Simulation.** Since current ML models for CG force fields suffer from high computational cost at every integration time step, Scherer et al.[14] proposed a kernel-based ML of the CG model for efficient simulations of molecular liquids. The central idea is to utilize a kernel machine to learn the mapping

$$Q \rightarrow F \tag{6}$$

where $Q$ is the representation of the systems, i.e., transformation of the coordinates of the atoms and $F$ is the force. The kernel function is $\hat{K} = K(Q_i, Q_j)$. Directly learning this mapping is very challenging, due to the large interpolation size of $Q$, even within a region defined by the cutoff distance $r_{cut}$. In this work, the $Q$ is decomposed into two terms $q^{(2)}$ and $q^{(3)}$, which represent two-body pair and three-body triplet interactions, respectively. To predict a rotationally invariant property, the two-body pair representation is chosen as the interparticle distance: $q^{(2)} = r_{m_{ab}}$, and the three-body triplet is chosen as a vector of three interparticle distances: $q^{(3)} = (r_{m_{ab}}, r_{m_{ac}}, r_{m_{bc}})^T$ (see the interaction $m$ part in Figure 7a). Accordingly, the force $F$ can be split into 2 terms $f^{(2)}$ and $f^{(3)}$, respectively (see the body expansion in Figure 7a). Therefore, the problem is simplified as the learning of the local mapping: $q^{(b)} \rightarrow f^{(b)}$, $b = 2, 3$, with the local kernel $\hat{k}^{(b)}$. Once trained for the local kernel, the ML model can be used to predict the local interactions, $f^*$. And after correlating the local kernel $\hat{k}^{(b)}$ to the global kernel $\hat{K}$, the new ML model can be used to predict the global force $F^*$.

One of the key aspects in this ML framework is to construct the kernel matrix $\hat{K}$. The basic energy conservation requires the kernel matrix is curl-free and the property rotates with the sample. Based on these requirements, three approaches are proposed: explicit rotations, integration over SO(3), and Hessian kernel. The performances of these three types of kernels are compared by studying the learning process for a two-body Lennard-Jones fluid (Figure 7b) and three-body Stillinger−Weber (SW) fluid (Figure 7c). It is found that, for the two-body interaction, there is no difference between the performances of the three kernels. However, for the three-body SW fluid, the Hessian kernel outperforms the other two kernels. Besides, by sorting and permutating the triplet representation vector $q^{(3)}$, the learning process will be significantly accelerated. Before performing MD simulations using the trained ML model, a switch function is used close to the cutoff distance to avoid numerical instabilities and maintain energy conservation. Furthermore, to reduce the error induced by the kernel predictions, the training data size is increased by the covariant meshing technique.

To demonstrate this ML framework, the trained CG model is adopted to run MD simulations. The results of RDF and ADF are compared with those from the referenced SW model. It is found all the curves are almost identical, which means the kernel predictions lead to the correct sampling of the canonical ensemble. In addition, the computational costs of these simulations are comparable with the original SW potential, without the restriction on the explicit functional form of three-body potentials.

This ML framework is further applied to the learning of the CG force field. After the CG process $R_i = R_i(r_j)$, the instantaneous collective forces (ICFs) are expressed as $F_i(r) = \sum_{j \in i} f_j(r)$, where $f_j(r)$ are the atomistic forces and $i$ is the CG bead. Here, the mapping is not directly from the $R$ to mean force. Instead, a two-step procedure is provided: first, the two-body force $F_i^{2\text{-body}}$ is obtained by the force-matching scheme, and second, the three-body forces are averaged as the residual force between target and two-body force: $\Delta F_i = F_i - F_i^{2\text{-body}}$. To learn these residual forces, a binned three-body Hessian kernel is adopted. The simulation results for liquid water with three different CG models are compared in Figure 7e. The Hessian kernel-based ML model is found to be very close to the atomistic results of RDF and ADF. More interestingly, it is found the ML model is even more accurate than the force-matched SW model.

This kernel-based ML framework demonstrates that by adding of switch function close to the cutoff distance and covariant meshing of the training data set, it can accurately capture two-body and three-body interactions with the comparative computational cost as the original SW model. And the extension to CG liquids confirms it is a promising technique to construct efficient two- and three-body models for a wide range of CG applications. However, for many-body interactions, i.e., four-body (dihedral) interaction, more complex and efficient kernel functions should be introduced. And the computational efficiency could be a significant challenge, as more interactions are included in the kernel matrix within the cutoff distance.


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

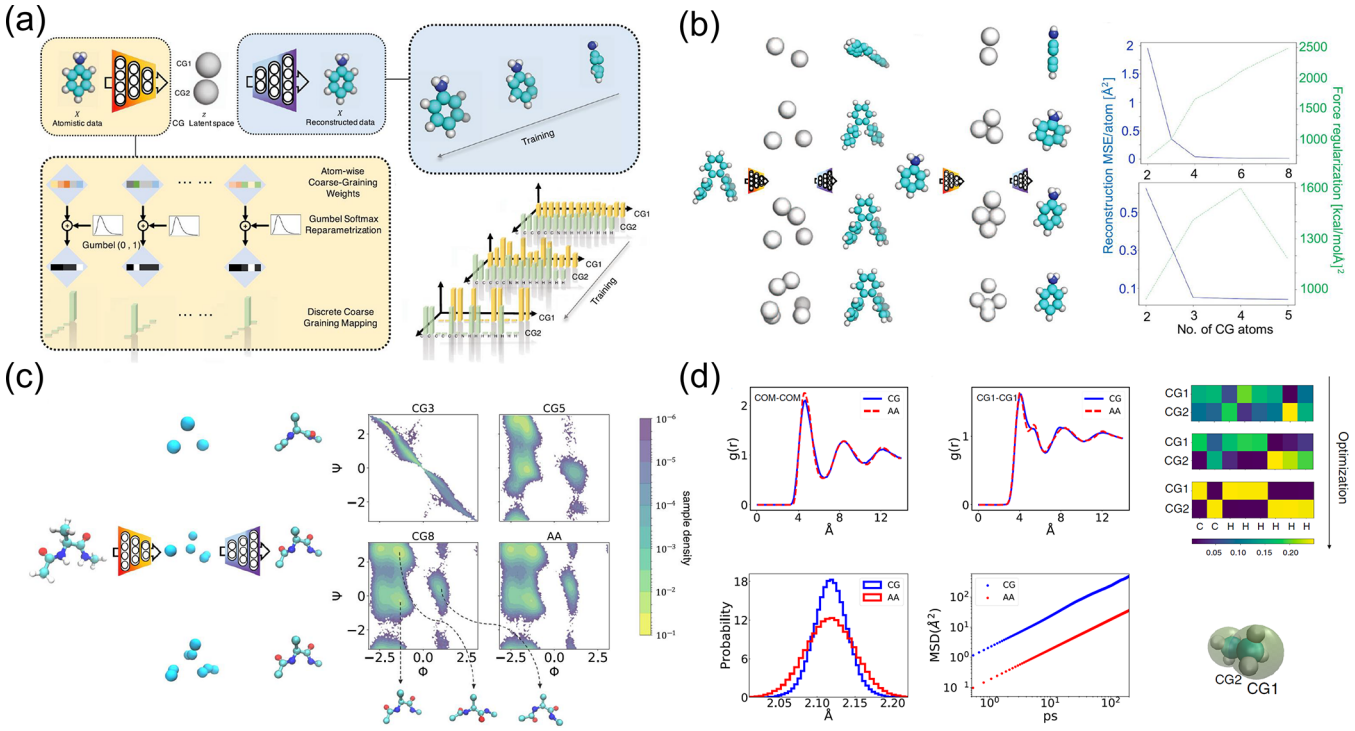



**Figure 8.** (a) CG autoencoding framework including the encoder, decoder, and the reconstruction of original all-atom data. (b) Reconstruction and mean force losses in Auto-Encoder training of gas-phase molecules with different resolutions. (c) Coarse-graining encoding and decoding for alanine dipeptide and the free energy profiles with different CG resolutions, compared to the atomistic simulation. (d) Comparison of the simulation results on RDF, bond distance distribution, and mean squared displacement (MSD) between atomistic and CG models for liquid ethane using a CG resolution of 2 per molecule (Reprinted with permission from the work of Wang et al.[15] Copyright 2019 Springer Nature Limited).

### 3.4. Autoencoders for CG Molecular Models.

The selection of an appropriate mapping from AA to CG models plays a critical role in developing CG models to reproduce consistent dynamics, structural correlation, and thermodynamics.[5] In general, the criteria for selecting CG mappings highly depends on the prior physical and chemical knowledge and intuition. Although many efforts have been devoted to develop forward- and back-mapping algorithms, the statistical connections are missing to reversibly bridge resolutions across different scales. To solve this issue, the powerful unsupervised learning technique—variational autoencoders (VAEs) are introduced. The VAEs compress the data through an information bottleneck that can first continuously map complex data into a low-dimensional latent space; second probabilistically infer the real data distribution via a generating process.[22] Accordingly, Wang et al.[15] proposed an autoencoder-based generative modeling framework that includes four steps: (1) learning discrete CG variables in latent space and decoding back to atomistic detail via geometric back—mapping; (2) using a reconstruction loss to help capture collective features from AA reference data; (3) regularizing the CG space with a semisupervised mean instantaneous force minimization to obtain a smoothed CG free-energy landscape; and (4) variationally finding the highly complex CG potential that matches the instantaneous mean force acting on the AA training data. Figure 8a shows the autoencoding framework. The atomistic data are first reconstructed by encoding atomistic trajectories through a low-dimensional bottleneck, and the CG mapping is parametrized by using the Gumbel-softmax reparametrization. In the training process, the encoder and decoder are variationally optimized to minimize the reconstruction loss:

$$L_{ae} = \frac{1}{N}\Sigma_{x\approx P(x)}[(D(E(x)) - x)^2 + \rho_h F_{inst}(E(x))^2] \quad (7)$$

where the first term on the right-hand side is atom-wise reconstruction loss and the second term is the average instantaneous mean force regularization. $\rho_h$ is the hyperparameter. Using this loss function, the force fluctuations of the encoded space is minimized by the instantaneous force regularizer and the CG free-energy landscape will be smoothed.

This unsupervised autoencoding process is first shown for gas-phase ortho-terphenyl (OTP) (Figure 8b) and aniline (Figure 8c). In the case of OTP, it is found if the number of degrees of freedom is less than 4, for example, 3-beads mapping that partitions each of the phenyl rings, this encoding loses the configuration information that describes the relative rotation of the two side rings, and thus leads to higher error for decoded structures. In the case of a small peptide molecule, the latent variables of 8 CG beads are adequate to recover heavy atom positions through the arrangement of hydrogen atoms is not fully reproduced.

With the minimization of the average of instantaneous mean forces as a regularization term, the learning of a smooth CG free energy surface is realized. This applicability of the framework is demonstrated by bulk simulations of liquids. Here, an example of liquid ethane is shown in Figure 8d. The CG resolution is 2. An autoencoder is trained to obtain the latent CG variables, and then a neural network-based CG force field is minimized by a force-matching scheme. Finally, the CG simulations are performed at the same density and temperature as the atomistic simulation. From the Figure 8d, it is found that the RDFs of both COM—COM and CG1—CG1 are captured


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

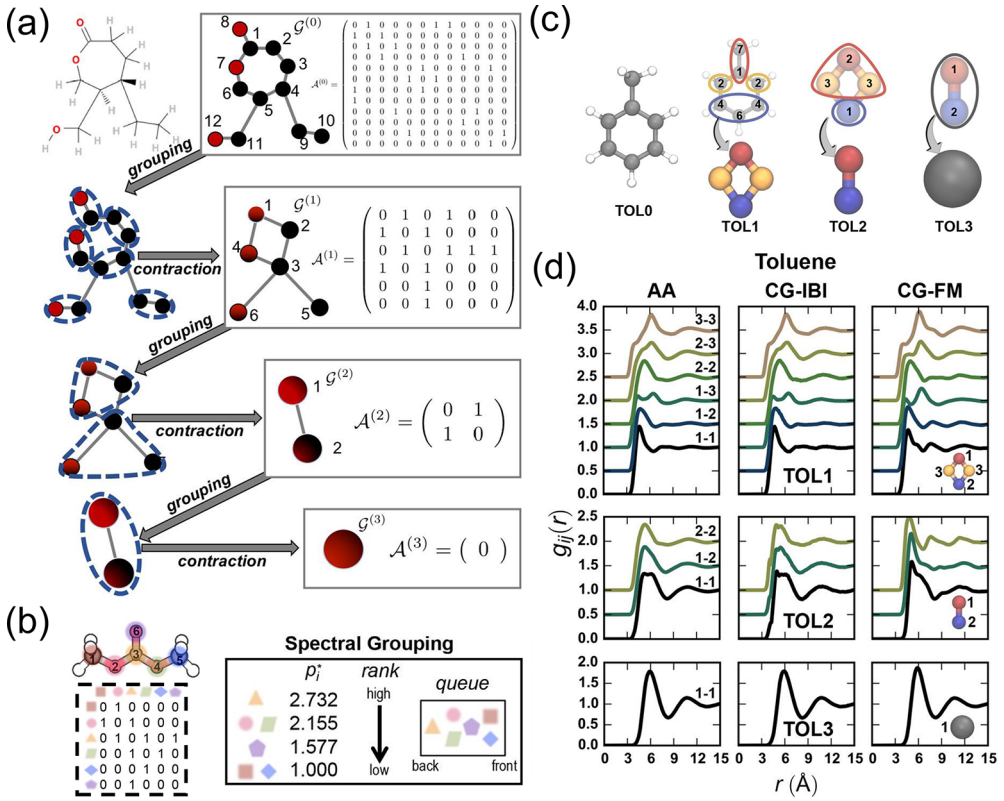



**Figure 9.** CG mapping scheme based on the graphic description of organic molecules. (a) Logistic of the graph-based coarse-graining (GBCG). The groupings and contractions produce different levels of CG descriptions of the original molecule. (b) (dashed-line box) Adjacency matrix A. (solid-line box) Grouping method contracts the adjacency matrix A. The spectral grouping ensures that those topologically important atoms are coarse-grained later than those less important atoms. (c) Example of toluene. Sequential CG mappings are shown. (d) Structural properties predicted by the CG models with potentials developed based on the different levels of coarse-graining. The results suggest the graphic CG mappings preserve the structural properties well (Reproduced from the work of Webb et al.[16] Copyright 2019 American Chemical Society).

accurately compared to those in the atomistic simulations. Also, the bond length distributions are in good agreement with that from atomistic simulations. The dynamic property informed from the mean squared displacement (MSD) is also shown in Figure 8d. The CG model exhibits faster dynamics, in comparison to atomistic simulations, which is the typical issue in classical CG models.[5]

This ML framework proposes to use the CG coordinates as latent variables that can be regularized with force. Through the training of encoding mapping, deterministic decoding, and a CG potential, a larger system is able to be simulated for a longer time, thus accelerating the MD simulations. However, this framework has its limitations. First, the deterministic CG mapping leads to an irreversible loss of information that is reflected in the reconstruction of average AA molecular structures. Second, due to the force-matching scheme, the individual pair correlation cannot be fully recovered compared to the atomistic trajectories. Lastly, as the same as other bottom-up approaches based on force-matching, this framework can only reproduce structural correlation function at one point in the thermodynamics space. It is not transferable among different thermodynamic conditions, which should be further addressed.[18]

**3.5. Graph-Based CG Molecular Model.** The representability, in terms of describing the ground-truth system, has drawn less attention compared to the transferability. However, its importance still deserves special notice because it is the baseline of building CG models and the success of a CG model relies on a good representation.[5] Webb et al.[16] proposed a graph-based coarse-graining (GBCG) scheme. The method essentially uses graph theory to describe the chemical connectivity of an organic molecule, mapping the ground-truth system of the molecule to a coarser description by basic graph operation of edge contraction, with a controllable degree of the coarse-graining.

Generally, a unique adjacency matrix is a complete description of the chemical structure of an organic molecule. The specific adjacency matrix includes vertexes, representing the atoms in the molecule, and edges, standing for the topological connections of these atoms. With this knowledge, the GBCG is nothing but a series of grouping and contraction operations, as illustrated in Figure 9a. In the grouping process, related vertexes are assigned to new groups; while in the contraction, vertexes in the new groups are combined together to form coarser sites. Here, a molecule with the SMILES string $[C(=O)1OCC(CO)C(CC)C1]$ is coarse-grained into models with 12, 6, 2, and 1 sites in a sequential and systematic manner.

The essential questions needed to be answered in the GBCG are how and in what sequence one should combine the vertexes. There are generally speaking different methods. However, in the following, a simple protocol, spectral grouping that uses only the adjacency matrix, is adopted for a simple but effective demonstration. An example of dimethyl carbonate is shown in Figure 9b. All the hydrogen atoms are first discarded and the rest are in united-atoms fashion, for simplicity. The 6 united-atoms are defined as 6 vertexes. A 6 × 6 symmetric


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

adjacency matrix $A$, highlighted in the dash-line box, is used to describe the connectivity of the system with the elements $A_{ij}$ equal to either 1 or 0, representing the connection between vertexes $N_i$ and $N_i$ exists or not, respectively. The grouping logic is in the solid-line box. The rank-order is defined by the eigenvector corresponding to the largest eigenvalues of the adjacency matrix $A$, based on the eigenvector centrality that counts the contributions of the vertexes to the overall connectivity. The grouping follows the queue, in which vertexes with a low-rank order lie in the front. Every vertex is grouped with its direct neighbor. The contraction of the atoms here is by the center of mass for simplicity. According to the listed ranks, the carbon atoms 1 and 5 are merged with their neighbors—the oxygen atoms 2 and 4 first. Then the oxygen atom 6 is combined with its neighbor—the carbon atom 3. Following this protocol, a toluene molecule is coarse-grained as shown in Figure 9c. The original molecule is labeled as TOL0, and it is sequentially coarse-grained into CG models TOL1, TOL2, and TOL3 that have 4, 2, and 1 beads, respectively. Validation to the example is shown in Figure 9d, where RDFs of the CG models at three different levels, equipped with potentials derived from IBI and force-matching methods, are shown. These results suggest that the graph method gives legitimate CG mappings that capture the essential structural properties, in comparison to the ground-truth atomistic system.

It is worth noting that the difference between the GBCG from the intuitive CG site definition as the center-of-mass of the original molecule is that the mappings in this method are in a sequence, corresponding to the topological connection of the chemical structure. Consequently, the GBCG gives a systematic way to derive a CG mapping with chemical essence.

## 4. DISCUSSION AND REMARKS

CG molecular simulations offer a unique opportunity to address challenging problems, such as phase separation, self-assembly, and crystallization of organic molecules and polymers. However, the classical CG models are limited by thermodynamic transferability and consistency.[5] Effective potential functions for the CG model are usually derived and optimized from one set of thermodynamic conditions. Thus, CG models derived from one thermodynamic state cannot be transferable to another set of conditions, limiting their applications to a small range of thermodynamic states. During the CG process, the free-energy landscape of the system is dramatically changed. Thus, it leads to thermodynamic inconsistency between AA and CG models. To overcome these limitations, the ML-based CG models provide an alternative solution, which can be optimized by using an extensive training data set of forces and structures from AA molecular simulations. In direct comparison with classical CG models, we believe that the ML-based CG models could have the following advantages: (i) thermodynamic consistency as neural networks can in principle approximate any function to arbitrary accuracy, which captures the surface roughness and highly nonconvex free-energy landscape of CG molecules; (ii) temperature/pressure transferability as neural networks has multiple layers of nonlinear functions and many model parameters, which can be optimized by extensive training data from AA simulations at different thermodynamic states; (iii) representability as neural networks can take two- and multibody molecular descriptors as inputs, which represents many-body interactions that are typically ignored in classical CG models. We should emphasize that the ML of AA and CG models tends to represent potential-energy and free-energy surfaces, respectively, by high-dimensional neural networks. During the ML of the AA model, both energies and forces are obtained directly from DFT simulations, leading to a fast and accurate neural network construction for the potential-energy surface. However, for the ML of CG models, only structures and forces are given from AA molecular simulations. The free-energy surface is challenging to estimate. Thus, most machine-learned CG models are formulated based on the force-matching scheme.[12−15] Note that both structure and force distributions are related to the free-energy landscape of the CG model. Therefore, the structural distributions (e.g., RDF and ADF) should be used to train the ML-based CG models in the near future. We also expect that these ML-based CG models can unify both structure- and force-matching during the optimization of neural network parameters.

Analogous to the typical ML of AA models for potential-energy surface construction, the ML of a CG model includes four major steps: (1) generating an extensive training data set using AA molecular simulations at different thermodynamic states; (2) mapping from a AA model to a CG model and extracting target structure or force distributions; (3) transforming Cartesian coordinates of CG particles to a suitable set of input coordinates (descriptors, features or symmetry functions) for the training of neural networks; (4) defining the architecture of neural networks and formulating a fitting procedure to optimize the weights and minimize the loss function. Each step can dramatically affect the accuracy and efficiency of the machine-learned CG models, in terms of reproducing the correct structural or dynamical properties of AA molecular systems. In the following, we discuss the efforts and unsolved issues in each step.

**4.1. Generation of Training Data Set.** The ML is usually a highly nonlinear fitting process with enormous parameters to be determined, in order to minimize the loss functions of neural networks. Theoretically, the ML model can fit an arbitrary system with the assumptions that the training data set is representative, diverse enough and the computational resource is adequate. In reality, the training data cannot cover all thermodynamic conditions. And actually, we are only interested in a system within a specific regime of thermodynamic states. Nevertheless, the training data set in the specific regime should be representative and diverse, which means the training data set needs to cover all the typical behaviors happening within this regime in terms of different thermodynamic states (e.g., pressure and temperature). The lack of data set will result in wrong thermodynamic transferability and consistency. The CG model trained by the data set within a specific regime of temperature or pressure cannot predict the physical states outside the trained thermodynamic regime (no extrapolation and exploration). In general, the sampling of the training data within the interested regime should be uniform, except for the extreme events. For example, two CG water beads can approach each other closely. If this scenario is not included in the training data set, the trained ML model will produce unstable MD simulations when two CG beads are close to each other. Frequent extrapolations will be necessary to handle this situation, and it can generate unphysical results. Therefore, the training data set should include enough information about different events, particularly extreme events. An alternative approach to relieve the lack of representation or diversity of the training data set is applying physical constraints


https://dx.doi.org/10.1021/acsomega.0c05321
ACS Omega 2021, 6, 1758−1772

in the trained ML model, such as physics-informed ML. For instance, in the CGnet model,[13] an *a priori* energy form is adopted to regularize the motion of CG particles in the simulations when extrapolation is needed. This can ensure the correct physical behaviors near the boundary of the training data set, as shown in Figure 6b.

**4.2. Mapping from AA to CG Models: Representation.** As mentioned before, the CG process from atomistic coordinates to CG sites strongly depends on the understandings of physical problems and chemical intuition.[5] CG will degenerate the system's degree of freedom and lose some important information that only can be observable in the atomistic simulations. A poor CG representation will lead to the loss of more important information and cannot reproduce it, no matter how efficiently and accurately the CG model performs in the following steps. Typically, for the well-known systems, for example, water molecules and peptides, it is easy to define the CG site on the oxygen atom and the backbone carbon atom, respectively. However, for unfamiliar systems, the center of mass is the most commonly used definition of the CG site, which is straightforward at first glance. But, the ambiguous nature of this method may jeopardize its usefulness. For example, when coarse-graining a linear flexible polymer chain, a CG site based on the center of mass of several monomers could be an inferior choice because the thermodynamic fluctuation for the center-of-mass potentially leads to an undesirable variation of the CG site.[5] Therefore, rigorous methods that give consistent and effective definitions of the mapping from AA to CG models are particularly important. To address this issue, the VAE-based ML framework can learn discrete CG variables in the latent space and decode them back to atomistic detail with high accuracy.[15] Besides, the graph-based CG adopts the graphic theory to describe the chemical connectivity of an organic molecule.[16] It maps the ground-truth system of the molecule to a coarser description by basic graphic operation of edge contraction, which provides a promising and efficient way for mapping from AA to CG models.

**4.3. Inputs of ML Models: Descriptors, Features, or Symmetry Functions.** The potential energy of the CG model should be only dependent on the internal interactions and be invariant with respect to the translation, rotation, or permutation of the entire molecules.[8] ML is a numerically fitting method and the output of an ML model depends on the absolute value of the input. Thus, Eulerian coordinates of the CG particles are not suitable as the inputs for ML models. Accordingly, molecular descriptors, features, or symmetry functions should be used to train these ML models.[23] The most straightforward input for the ML-based CG model is the distance between two CG particles. However, this choice is not unique and can0not capture many-body interactions. Currently, there are different transformations of Eulerian coordinates of CG particles as inputs of ML models, such as using local coordinate systems,[12] two- or three-body correlation functions,[14] permutation-invariant distance metrics,[24] and VAE-based encoder-decoder.[15] Apart from the energy invariance, the symmetry functions for ML models should be first- and second-order continuous that the force can be derived as the gradient of CG potential with respect to coordinates. Furthermore, the symmetry functions should be broad enough to capture the interactions between two nearby CG particles and ensure the decay of energy to zero when the distance between two CG particles approaches the cutoff distance. Therefore, too simple or too small symmetry functions can lower the accuracy of machine-learned CG models.[8]

**4.4. Optimization of Weights in Neural Networks.** In the development of a machine-learned CG model, the optimization of parameters is always the most time-consuming part. For neural networks, a variety of gradient-based optimization algorithms are available, from simple gradient descent (back-propagation) and variants-like RPROP (resilient back-propagation) or Adam to higher-order methods, such as the L-BFGS method and extended Kalman filter (EKF).[25] The choice of optimization algorithms depends on the specific problem. For example, EKF performs very well in training one or two hidden layers with each having less than 100 neurons but poorly in training networks with four or more hidden layers. Also, considering the resources of HPC, we need to choose the optimization algorithm, which can perform training with a large data set more efficiently.

**4.5. Future Opportunities.** So far, the machine-learned CG models have made extensive efforts in either one or two aspects of the above discussions to improve accuracy. This has led to limited applications of these models. For example, the methods of kernel-based and graph-based CG models focus on the representation aspect and provide different protocols to retain the information contained in the molecular structure as much as possible. While the optimization of these CG models is limited to the basis of the force-matching scheme. It can lead to poor reproductions of structural properties, such as RDF, ADF, etc. A potential way to solve this issue is by adding another loss function (target), such as RDF, to optimize the CG models. Due to the complexity of high dimensional neural networks, both structural and force distributions could be simultaneously reproduced by the machine-learned CG models. In terms of the training data set, the ML-BOP model focuses on the temperature-transferability by covering a wide range of temperature states.[9] Meanwhile, pressure-transferability remains to be studied. If we include more train data sets at different pressures, the ML-BOP model may realize pressure-transferability and, then, can predict the nucleation process of the water molecules more accurately. For the choice of symmetry functions, in general, it depends on the experience of trial-and-error. On one side, the symmetry functions should be more than enough to capture all the typical interactions among CG particles. On the other side, symmetry functions should be small enough to have good performance in terms of computational efficiency. Therefore, we need to pay close attention to how to choose appropriate symmetry functions. In conclusion, we are still facing enormous challenges and opportunities for constructing an accurate and efficient machine-learned CG model to be thermodynamically consistent, transferable, and representative.

## AUTHOR INFORMATION


### Corresponding Author

**Ying Li** − *Department of Mechanical Engineering, University of Connecticut, Storrs, Connecticut 06269, United States; Polymer Program, Institute of Materials Science, University of Connecticut, Storrs, Connecticut 06269, United States;* orcid.org/0000-0002-1487-3350; Phone: +1 860 4867110; Email: yingli@engr.uconn.edu; Fax: +1 860 4865088




https://dx.doi.org/10.1021/acsomega.0c05321
*ACS Omega* 2021, 6, 1758−1772

### Authors

**Huilin Ye** − *Department of Mechanical Engineering, University of Connecticut, Storrs, Connecticut 06269, United States;* orcid.org/0000-0002-8041-4056

**Weikang Xian** − *Department of Mechanical Engineering, University of Connecticut, Storrs, Connecticut 06269, United States;* orcid.org/0000-0002-6802-651X

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.0c05321

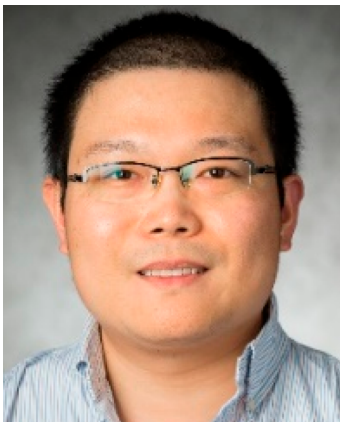

### Author Contributions

[§]H.Y. and W.X. contributed equally to this work.

### Notes

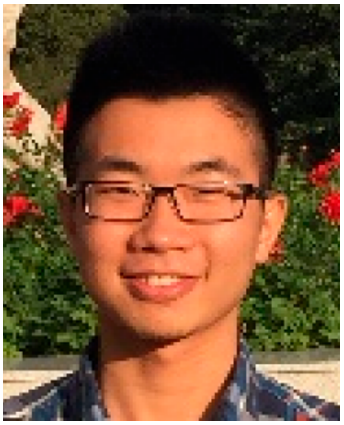
The authors declare no competing financial interest.

### Biographies



Dr. Ying Li joined the University of Connecticut in 2015 as an Assistant Professor in the Department of Mechanical Engineering. He received his Ph.D. in 2015 from Northwestern University, focusing on the multiscale modeling of soft matter and related biomedical applications. His current research interests are the following: multiscale modeling, computational materials design, mechanics and physics of polymers, machine learning-accelerated polymer design. Dr. Li's achievements in research have been widely recognized by fellowships and awards, including the Air Force's Young Investigator Award (2020), 3M Non-Tenured Faculty Award (2020), ASME Haythornthwaite Young Investigator Award (2019), NSF CRII Award (2018), and many others. He has authored and co-authored more than 90 peer-reviewed journal articles, including those published in *Physical Review Letters*, *ACS Nano*, *Biomaterials*, *Nanoscale*, *Macromolecules*, *Journal of Mechanics and Physics of Solids*, *Journal of Fluid Mechanics*, etc.

Huilin Ye is a Ph.D. candidate in Mechanical Engineering at the University of Connecticut. His research interest is mainly in developing high-fidelity computational methods in applications for bio- and mechanical engineering. The key tasks in this area include two aspects: (1) fluid−structure interaction algorithms (FSIAs) and (2) high performance computing (HPC). This novel numerical scheme has been successfully applied in a targeted drug delivery system for capturing the dynamic motion of micro- and nanoparticles in blood flow and machine learned coarse-grained modeling. Ye's works have been recognized by fellowships and awards including the Generic Electric Fellowship for Innovation, the best paper award of the FDTC student paper competition in EMI (2018) from the ASCE, and a non-academic research internship from the National Science Foundation.



Weikang Xian is a Ph.D. student in Mechanical Engineering at the University of Connecticut, under the supervision of Prof. Dr. Ying Li. His research interest is mainly on developing multiscale computational methods to study mechanical behaviors of polymers and metamaterials. He received his bachelor's degree from Jilin University, China. Before moving to University of Connecticut, he had been a visiting student in Mechanical and Aerospace Engineering at the University of California at Los Angeles working with Prof. Dr. Lihua Jin.

## ■ ACKNOWLEDGMENTS

Y.L. acknowledges the financial support from the Air Force Office of Scientific Research (FA9550-20-1-0183; Program manager: Dr. Ming-Jen Pan) and the National Science Foundation (CMMI-1934829). H.Y., W.X., and Y.L. are all grateful for the support from the Department of Mechanical Engineering at the University of Connecticut (UConn). H.Y. was partially supported by a fellowship grant from GE's Industrial Solutions Business Unit under a GE−UConn partnership agreement. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Industrial Solutions or UConn. This research benefited in part from the computational resources and staff contributions provided by the Booth Engineering Center for Advanced Technology (BECAT) at UConn.

## ■ REFERENCES

(1) Buch, I.; Harvey, M. J.; Giorgino, T.; Anderson, D. P.; De Fabritiis, G. High-throughput all-atom molecular dynamics simulations using distributed computing. *J. Chem. Inf. Model.* **2010**, *50*, 397−403.

(2) Noid, W. G. Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **2013**, *139*, 090901.

(3) Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C. The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008**, *128*, 244114.
(4) Huang, D. M.; Faller, R.; Do, K.; Moulé, A. J. Coarse-grained computer simulations of polymer/fullerene bulk heterojunctions for organic photovoltaic applications. *J. Chem. Theory Comput.* **2010**, *6*, 526−537.
(5) Li, Y.; Abberton, B. C.; Kröger, M.; Liu, W. K. Challenges in multiscale modeling of polymer dynamics. *Polymers* **2013**, *5*, 751−832.
(6) Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; De Vries, A. H. The MARTINI force field: coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **2007**, *111*, 7812−7824.
(7) Zhang, Z.; Lu, L.; Noid, W. G.; Krishna, V.; Pfaendtner, J.; Voth, G. A. A systematic methodology for defining coarse-grained sites in large biomolecules. *Biophys. J.* **2008**, *95*, 5073−5083.
(8) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.
(9) Chan, H.; Cherukara, M. J.; Narayanan, B.; Loeffler, T. D.; Benmore, C.; Gray, S. K.; Sankaranarayanan, S. K. Machine learning coarse grained models for water. *Nat. Commun.* **2019**, *10*, 1−14.
(10) Moradzadeh, A.; Aluru, N. R. Transfer-Learning-Based coarse-graining method for Simple fluids: Toward Deep Inverse liquid-state theory. *J. Phys. Chem. Lett.* **2019**, *10*, 1242−1250.
(11) Bejagam, K. K.; Singh, S.; An, Y.; Deshmukh, S. A. Machine-learned coarse-grained models. *J. Phys. Chem. Lett.* **2018**, *9*, 4667−4672.
(12) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. DeePCG: Constructing coarse-grained models via deep neural networks. *J. Chem. Phys.* **2018**, *149*, 034101.
(13) Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; De Fabritiis, G.; Noé, F.; Clementi, C. Machine learning of coarse-grained molecular dynamics force fields. *ACS Cent. Sci.* **2019**, *5*, 755−767.
(14) Scherer, C.; Scheid, R.; Andrienko, D.; Bereau, T. Kernel-based machine learning for efficient simulations of molecular liquids. *J. Chem. Theory Comput.* **2020**, *16*, 3194−3204.
(15) Wang, W.; Gómez-Bombarelli, R. Coarse-graining auto-encoders for molecular dynamics. *npj Comput. Mater.* **2019**, *5*, 1−9.
(16) Webb, M. A.; Delannoy, J.-Y.; De Pablo, J. J. Graph-based approach to systematic molecular coarse-graining. *J. Chem. Theory Comput.* **2019**, *15*, 1199−1208.
(17) John, S.; Csányi, G. Many-body coarse-grained interactions using Gaussian approximation potentials. *J. Phys. Chem. B* **2017**, *121*, 10934−10949.
(18) Ruza, J.; Wang, W.; Schwalbe-Koda, D.; Axelrod, S.; Harris, W. H.; Gómez-Bombarelli, R. Temperature-transferable coarse-graining of ionic liquids with dual graph convolutional neural networks. *J. Chem. Phys.* **2020**, *153*, 164501.
(19) Durumeric, A. E.; Voth, G. A. Adversarial-residual-coarse-graining: Applying machine learning theory to systematic molecular coarse-graining. *J. Chem. Phys.* **2019**, *151*, 124110.
(20) Mitchell, M. *An introduction to genetic algorithms*; MIT press, 1998.
(21) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **2018**, *120*, 143001.
(22) Tishby, N.; Zaslavsky, N. *Deep learning and the information bottleneck principle*; 2015 IEEE Information Theory Workshop (ITW), 2015; pp 1−5.
(23) Chen, G.; Shen, Z.; Iyer, A.; Ghumman, U. F.; Tang, S.; Bi, J.; Chen, W.; Li, Y. Machine-Learning-Assisted De Novo Design of Organic Molecules and Polymers: Opportunities and Challenges. *Polymers* **2020**, *12*, 163.
(24) Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; Von Lilienfeld, O. A.; Tkatchenko, A.; Muller, K.-R. Assessment and validation of machine learning methods for predicting molecular atomization energies. *J. Chem. Theory Comput.* **2013**, *9*, 3404−3419.
(25) Singraber, A.; Morawietz, T.; Behler, J.; Dellago, C. Parallel multistream training of high-dimensional neural network potentials. *J. Chem. Theory Comput.* **2019**, *15*, 3075−3092.


https://dx.doi.org/10.1021/acsomega.0c05321
*ACS Omega* 2021, 6, 1758−1772