Soft Matter

Data-driven coarse-grained modeling of polymers in solution with structural and dynamic properties conserved

Journal:	*Soft Matter*
Manuscript ID	SM-ART-06-2020-001019.R1
Article Type:	Paper
Date Submitted by the Author:	20-Jul-2020
Complete List of Authors:	Wang, Shu; University of Wisconsin-Madison, Mechanical Engineering Ma, Zhan; University of Wisconsin-Madison, Mechanical Engineering Pan, Wenxiao; University of Wisconsin Madison, Mechanical Engineering

Soft Matter

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Data-driven coarse-grained modeling of polymers in solution with structural and dynamic properties conserved

Shu Wang,[a] Zhan Ma,[a] and Wenxiao Pan[a*]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

We present data-driven coarse-grained (CG) modeling for polymers in solution, which conserves the dynamic as well as structural properties of the underlying atomistic system. The CG modeling is built upon the framework of generalized Langevin equation (GLE). The key is to determine each term in the GLE by directly linking it to atomistic data. In particular, we propose a two-stage Gaussian process-based Bayesian optimization method to infer the non-Markovian memory kernel from the data of velocity autocorrelation function (VACF). Considering that the long-time behaviors of VACF and memory kernel for polymer solutions can exhibit hydrodynamic scaling (algebraic decay with time), we further develop an active learning method to determine the emergence of hydrodynamic scaling, which can accelerate the inference process of memory kernel. The proposed methods do not rely on how the mean force or CG potential in the GLE is constructed. Thus, we also compare two methods for constructing the CG potential: a deep learning method and the iterative Boltzmann inversion method. With the memory kernel and CG potential determined, the GLE is mapped onto an extended Markovian process to circumvent the expensive cost of directly solving the GLE. The accuracy and computational efficiency of the proposed CG modeling are assessed in a model star-polymer solution system at three representative concentrations. By comparing with the reference atomistic simulation results, we demonstrate the proposed CG modeling can robustly and accurately reproduce the dynamic and structural properties of polymers in solution.

1 Introduction

Atomistic simulations via, e.g., all-atom molecular dynamics (MD), have been widely employed to simulate polymers and biomolecules in solution. By tracking individual atoms of molecules and solvent, these simulations are accurate yet expensive. For large-scale polymer solution systems, the mesoscopic properties and collective dynamics of polymers can be of more importance. Thus, it may not be necessary to simulate all atomistic details of the system, but instead eliminate or average out certain degrees of freedom (DOFs) properly to reduce simulation cost. It is so-called coarse-grained (CG) modeling [1–5]. The removal of highly-fluctuating atomic DOFs and the larger characteristic length scale of CG coordinates permit to employ larger time steps in CG simulations. For modeling polymers in solution, if not only the DOFs representing polymer molecules are reduced but also the solvent DOFs are eliminated, it leads to implicit-solvent CG modeling [6–9]. Significantly reduced DOFs and larger time steps would enable CG simulations much more efficient than full atomistic simulations, and hence more applicable to simulating large-scale polymer solution systems in practical applications [10–13]. However, the challenge of CG modeling is to conserve both the structural and dynamic properties of polymers in the coarse-graining process. In particular, if the solvent DOFs are eliminated, the solvent-mediated dynamic effect must be properly incorporated in CG modeling.

To conserve the structural properties (e.g., radial and angular distribution functions), the CG potential must be correctly constructed. To this end, the existing methods in literature include the iterative Boltzmann inversion (IBI) method [14], inverse Monte Carlo [15], force matching method [16], minimization of relative entropy [17,18], etc. These methods typically employ assumptions or approximations for the CG potential; for example, assume a two-body pairwise potential [14,15] or augment the pairwise potential with an additive function of local densities of CG sites [17,18] to approximate the many-body interactions between CG coordinates. In addition to these methods, machine learning techniques [19–21] were also employed to determine the CG potential, which generally can avoid employing *ad hoc* assumptions or approximations for the CG potential. All these efforts are limited to only conserving the structural properties of the underlying atomistic system.

To conserve the dynamic properties (e.g., velocity autocorrela-

[a] *Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706.*
[*] Corresponding author. E-mail: wpan9@wisc.edu

tion function (VACF) and diffusivity), the dynamic effect of unresolved DOFs (including solvent) must be properly accounted. To this end, a non-Markovian dynamics must be introduced in the CG modeling since elimination of DOFs results in a non-Markovian memory in the dynamics of CG variables, as discussed in literature[22–27] and also in our prior work[28]. Particularly in implicit-solvent CG modeling, the non-Markovian memory can play an important role in producing correct long-time dynamics[26,28–30]. The generalized Langevin equation (GLE) provides a theoretically sound framework for CG modeling to describe the non-Markovian dynamics of CG variables. However, efforts must be taken to properly construct the memory kernel (function) in the GLE such that the dynamics of the underlying atomistic system can be accurately reproduced by the GLE. In this regard, there have been different approaches developed in literature. Among them, most methods are not applicable to polymers in solution. For example, some methods[23,31,32] only concern polymers in melt but cannot consider the solvent-mediated effect on dynamics in polymer solutions. Our prior work[28] and some efforts in literature[24,30,33] neglect the CG potential (or mean force) in the GLE and hence cannot reproduce the structural properties of the reference atomistic system. Jung et al.[26,34] proposed an inverse iterative procedure to determine the memory kernel from the VACF data of atomistic simulations, which was applied in the context of nanocolloids in dilute solution. Given an appropriate initial guess, this method is effective to find a memory kernel for the GLE to produce the target VACF. However, the convergence of iterations can depend on the choice of initial guess. Also, directly solving the GLE is expensive due to storage of historical information of CG variables, numerical evaluation of convolution, and generation of colored noise. This method cannot alleviate this expensive cost. Their results only showed short-time dynamics reproduced by the GLE with the constructed memory kernel.

In this paper, we propose a different approach to infer the memory kernel from the VACF data. The key idea is a two-stage Gaussian process-based Bayesian optimization. The memory kernel is approximated by a truncated expansion of exponentially damped oscillators. The parameters in the expansion are optimized via the two-stage Bayesian optimization with the objective function associated with the VACF. The optimization process allows for efficient use of data with maximum information gain via adaptive sampling guided by Gaussian process. The memory kernel inferred by such permits to map the GLE to a Markovian process extended in a higher dimensional space, which is much cheaper to solve than the GLE and hence leads to more efficient CG simulations. In addition, we consider the fact that the long-time behaviors of VACF and memory kernel for polymer solutions can exhibit hydrodynamic scaling or algebraic decay with time[35–38]. Thus, we propose an active learning process to determine the emergence of hydrodynamic scaling, which can accelerate the inference process of memory kernel. These proposed methods should not rely on how the CG potential is constructed. Thus, we employ two different methods for constructing the CG potential to conserve the structural properties of the underlying atomistic system: one is a deep learning method , i.e. a deep neural networks (DNN) method; and the other is the IBI method[14] commonly employed in CG modeling of polymers. Using a model polymer solution system, we demonstrate the accuracy and efficiency of the proposed CG modeling in reproducing both structural and dynamic properties of polymers in solution, compared with the reference atomistic simulation results.

The rest of the paper is organized as follows. In §2, we describe the theoretical framework and GLE, based on which the CG modeling is established. In §3.1, we briefly describe the DNN and IBI methods employed for constructing the CG potential. §3.2 explains in detail the new method proposed for inference of memory kernel from the data of VACF obtained in atomistic simulations, which consists of the two-stage Bayesian optimization and an automated, active learning process for detection of algebraic decay in VACF. The equivalence of the GLE and the extended Markovian process is explained in §3.3. We present all results in §4, where a benchmark, star-polymer solution system at different concentrations is studied to assess the accuracy and computational cost of the CG simulations. Finally, we conclude and summarize our main findings and contributions in §5.

2 Theoretical background

Without loss of generality, the atomistic system consists of n atoms (beads) in polymer molecules, with coordinates $\mathbf{r} = \{\mathbf{r}_i | i = 1,2,\ldots,n\}$ and momenta $\mathbf{p} = \{\mathbf{p}_i | i = 1,2,\ldots,n\}$. In coarse-graining, n atoms are coarse-grained as N clusters (referred to as CG particles), and each cluster contains n_c atoms. The CG particles' positions $\mathbf{R} = \{\mathbf{R}_I | I = 1,2,\ldots,N\}$ and momenta $\mathbf{P} = \{\mathbf{P}_I | I = 1,2,\ldots,N\}$ constitute an extensive set of CG variables of the N-body CG system.

To be consistent in notation, we use the lowercase m_i, $\mathbf{r}_i$, and $\mathbf{p}_i$ to represent the mass, position, and momentum of the i-th atom in the atomistic system; and the uppercase M_I, $\mathbf{R}_I$, and $\mathbf{P}_I$ denote the mass, position, and momentum of the I-th CG particle in the CG system. The variables of the atomistic and CG systems are related via:

$$M_I = \sum_{i=1}^{n_c} m_{Ii}\,, \quad \mathbf{R}_I = \frac{1}{M_I}\sum_{i=1}^{n_c} m_{Ii}\mathbf{r}_{Ii}\,, \quad \mathbf{P}_I = \sum_{i=1}^{n_c} \mathbf{p}_{Ii}\,, \tag{1}$$

where m_{Ii} is the mass of the i-th atom in the I-th CG particle; and $\mathbf{R}_I$ and $\mathbf{P}_I$ are defined as the center-of-mass (COM) position and total momentum of the I-th cluster, respectively.

The dynamics of the CG system is governed by the GLE as:

$$\dot{\mathbf{P}}_I = \langle \mathbf{F}_I \rangle - \int_0^t K(t-t') M_I^{-1} \mathbf{P}_I(t') dt' + \tilde{\mathbf{F}}_I\,. \tag{2}$$

In the right-hand side of Eq. (2), the first term $\langle \mathbf{F}_I \rangle = -\nabla_{\mathbf{R}_I} U$ is the ensemble-average mean force with $U = \sum_{I=1}^{N} U_I$, where U_I is the potential of mean force (PMF) or referred to as the CG potential. The third term $\tilde{\mathbf{F}}_I$ denotes the random force. The second term (referred to as the dissipative force) has a memory kernel $K(t-t')$, which is related to the random force by: $K(t) = (1/k_B T)\langle [\tilde{\mathbf{F}}_I(t)]^{\intercal} [\tilde{\mathbf{F}}_I(0)] \rangle$ with Boltzmann constant k_B and thermodynamic temperature T to satisfy the second fluctuation-dissipation theorem[39]. The dissipative and random forces compensate for the lost atomic DOFs that result from coarse-graining.

To conserve both structural and dynamic properties of the underlying atomistic system, the CG potential and memory kernel in Eq. (2) must be directly linked to the atomistic system.

3 Methodology

In this section, we discuss how to construct the CG potential and memory kernel from atomistic data.

3.1 Construction of CG potential

The structural properties of the CG system are mainly determined by the CG potential that in general is nonlinear and can consist of many-body interactions. Without *ad hoc* approximations such as limiting the potential to two-body (or pairwise) contributions, a DNN method can be employed to construct the CG potential. It is adapted from the method proposed by Zhang et al.[20], where the CG coordinates are at one atom of water molecule. In this work, the CG coordinates are at the COM of polymers. The performance of the DNN method is compared with the IBI method[14] commonly employed in CG modeling of polymers.

3.1.1 Deep learning

In the deep learning method, a DNN representation $U_I^{\boldsymbol{\omega}}$ for the CG potential (or PMF) $U_I(\mathbf{R})$ is sought with $\boldsymbol{\omega}$ the network parameters. To ensure the output $U_I^{\boldsymbol{\omega}}$ preserves the translational, rotational, and permutational symmetries of the CG free energy surface, the input of the DNN is required to preserve the same symmetries. To this end, the global coordinates $\mathbf{R}$ of the CG system are transformed into a local descriptor matrix $\mathbf{D}_I$ as the input of DNN, which describes the configuration of neighbors of I-th CG particle in its local Cartesian coordinates.

Following the work of Zhang et al.[20,40], local coordinates of the I-th CG particle are first constructed based on the positions of the I-th CG particle and its first and second nearest neighbors. The origin is set at $\mathbf{R}_I$. In the local coordinates, $\{\bar{x}_{IJ}, \bar{y}_{IJ}, \bar{z}_{IJ}\}$ defines the vector from I to J. Then, $\mathbf{D}_I$ takes the following form:

$$\begin{aligned} \mathbf{D}_I &= \{\mathbf{D}_{IJ} | J \in \{\text{neighbors of } I \text{ that satisfy } R_{IJ} \le R_{cut}\}\} \\ \mathbf{D}_{IJ} &= \{\frac{1}{R_{IJ}}, \frac{\bar{x}_{IJ}}{R_{IJ}^2}, \frac{\bar{y}_{IJ}}{R_{IJ}^2}, \frac{\bar{z}_{IJ}}{R_{IJ}^2}\}, \end{aligned} \tag{3}$$

where R_{cut} is the cut-off radius; $R_{IJ} = |\mathbf{R}_{IJ}|$ with $\mathbf{R}_{IJ} = \mathbf{R}_I - \mathbf{R}_J$; and $\mathbf{D}_{IJ}$ is sorted in $\mathbf{D}_I$ with ascending R_{IJ}.

With the input $\mathbf{D}_I$ and output U_I, the architecture of the DNN is illustrated in Fig. 1. To train the DNN, the loss function used in the training process is:

$$L(\boldsymbol{\omega}) = \frac{1}{\Upsilon N} \sum_{\upsilon=1}^{\Upsilon} \sum_{I=1}^{N} |\mathbf{F}_I(\mathbf{R}^\upsilon) + \nabla_{\mathbf{R}_I} U^{\boldsymbol{\omega}}(\mathbf{R}^\upsilon)|^2, \tag{4}$$

where υ is the υ-th configuration; Υ is the total number of configurations used in the training process; $U^{\boldsymbol{\omega}}(\mathbf{R}^\upsilon) = \sum_{I=1}^{N} U_I^{\boldsymbol{\omega}}(\mathbf{R}^\upsilon)$; and $\mathbf{F}_I$ is the instantaneous total force on the I-th CG particle. Here, we assume the CG particles $I = 1, 2, \ldots, N$ have the same DNN model of CG potential. In this work, the local descriptor matrix $\mathbf{D}_I$ was generated through the DeePMD-kit package[40], which also provides the interface to TensorFlow for training the DNN and calling the trained DNN model to calculate $U_I^{\boldsymbol{\omega}}(\mathbf{R})$.

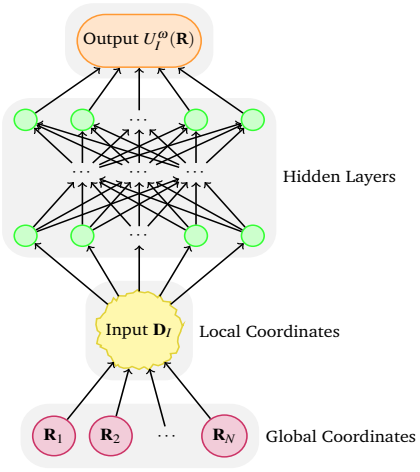

Fig. 1 Schematic of the feedforward DNN architecture to learn the CG potential.

The specific architecture of the DNN for each system was determined according to the following theoretical and empirical rules[41,42]. 1) An inverted pyramid architecture with the number of nodes (or neurons) per hidden layer decreasing from the inner layer to the outer layer is adopted to be compatible with the fact that the number of inputs is much larger than the number of outputs. In the present work, there are averagely 80 inputs and 1 output in all cases considered. 2) Since the training cost increases as the numbers of nodes and layers increase, the neural network with fewer nodes and layers is preferred. 3) Assuming the same total number of nodes, having more hidden layers is more effective than putting more nodes per layer to represent a more complex function between the outputs and inputs. 4) The number of nodes per hidden layer should be between the numbers of inputs and outputs. 5) When adding more nodes or a hidden layer cannot further reduce the loss function, the architecture of the neural network is considered optimal.

3.1.2 Iterative Boltzmann inversion

The IBI method assumes the CG potential is pairwise: $U = \sum_{I,J} U(R_{IJ})$, and constructs the pair potential $U(R_{IJ})$ via an inverse iterative process by reproducing the RDF of the reference atomistic system[14]. The iteration follows[14,43]:

$$U^{i+1}(R_{IJ}) = U^i(R_{IJ}) - \eta k_B T \ln\left(\frac{\mathrm{RDF}_{CG}^i(R_{IJ})}{\mathrm{RDF}_A(R_{IJ})}\right), \tag{5}$$

where the superscript i denotes the i-th iteration; $\eta < 1$ is a scaling factor that helps improve the convergence and stability of the iteration process[43]; and $\mathrm{RDF}_A(R_{IJ})$ is the RDF of the reference atomistic system. The initial guess for the iteration in Eq. (5) is given by[14]:

$$U^0(R_{IJ}) = -k_B T \ln[\mathrm{RDF}_A(R_{IJ})]. \tag{6}$$

The iteration in Eq. (5) is terminated once $\frac{\|\mathrm{RDF}_{CG}(R_{IJ}) - \mathrm{RDF}_A(R_{IJ})\|_2}{\|\mathrm{RDF}_A(R_{IJ})\|_2}$ is no greater than a tolerance, which was set as 0.02 in this work. Here, $\|\cdot\|_2$ denotes the l-2 norm of discrete data. The VOTCA

package[44] was employed to implement the IBI method in the present work.

3.2 Inference of memory kernel

To conserve the dynamics of the underlying atomistic system in CG modeling, the memory kernel K in Eq. (2) must be properly determined. In general cases of N-body ($N > 1$) CG systems, the total force on each body has contributions from others. To reproduce the structural properties, we need to consider the mean force $\langle \mathbf{F}_I \rangle$ in Eq. (2) as a function of $\mathbf{R}$, as discussed in §3.1. If we regard $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_N\}$ as a point in the CG phase space, $\langle \mathbf{F}_I \rangle$ is associated with the normalized partition function of all atomistic configurations at phase point $\mathbf{R}$. As a result, if using a forward approach to determine the memory kernel, as in our prior work[28] or others' work[27,30], the memory kernel must be also a function of $\mathbf{R}$. However, how the memory kernel depends on $\mathbf{R}$ cannot be assumed *a priori*. Determining the memory kernel as a function of both t and $\mathbf{R}$ is challenging. Thus, previous work either only considered one-body CG system[27], or neglected to reproduce the structural properties by regarding $\langle F_I \rangle$ as the average over all phase points and hence taking $\langle \mathbf{F}_I \rangle = 0$ in the GLE[28,30], or assumed t and $\mathbf{R}$ were separable and the function of $\mathbf{R}$ was only dependent on $\mathbf{R}_{IJ}$ (pairwise contributions from neighbors)[23].

To alleviate the difficulty in determining the memory kernel in the meanwhile being able to reproduce the structural properties, we propose in this work a new approach. First, the memory kernel in Eq. (2) is constructed as the average over all phase points and hence is only a function of time t. The memory kernel $K(t)$ is then inferred via an inverse optimization process from the atomistic data of VACF. This approach does not rely on what the mean force (or CG potential) is and how it is constructed. To proceed, $K(t)$ is first approximated by an asymptotic expansion as:

$$K(t) = \sum_{l=1}^{\mathscr{N}} \exp(-\frac{a_l}{2}t)[b_l \cos(q_l t) + c_l \sin(q_l t)] \,, \tag{7}$$

where $\{a_l, b_l, c_l, q_l\} \in \boldsymbol{\lambda}$ (parameter space). Given Eq. (7) truncated to finite terms, an optimization problem in $\boldsymbol{\lambda}$ space can be framed to determine the parameters $\{a_l, b_l, c_l, q_l\} \in \boldsymbol{\lambda}$ that minimize the difference on $\text{VACF}(t) = \langle V(t)V(0) \rangle$ of the CG system with respect to the reference atomistic system:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda}}{\text{argmin}}\, \Pi(\boldsymbol{\lambda}) \,, \tag{8}$$

where $\Pi(\boldsymbol{\lambda})$ is the objective function and defined as:

$$\Pi(\boldsymbol{\lambda}) = \frac{\| \langle \text{VACF}_{CG}(\boldsymbol{\lambda}) \rangle - \langle \text{VACF}_A \rangle \|_2}{\| \langle \text{VACF}_A \rangle \|_2} \,. \tag{9}$$

Here, the VACF is chosen for optimization because the behaviors of VACF imply all dynamic properties of the system[45]. The data of $\text{VACF}_{CG}(\boldsymbol{\lambda})$ and VACF_A are attained from the CG and atomistic simulations, respectively. We aim to determine $K(t)$ up to $t \leq t_{\text{cut}}$ since for $t > t_{\text{cut}}$, $|\text{VACF}(t)/\text{VACF}(0)| \leq 10^{-3}$, and the dynamic properties are considered nearly invariant.

3.2.1 Gaussian process (GP)-based Bayesian optimization

Solving the optimization problem in Eq. (8) is challenging due to: *i*) the objective Π is a black box for which no closed form is known nor its gradient and Hessian; *ii*) Π is expensive to evaluate; *iii*) evaluations of Π are noisy; and, *iv*) $\boldsymbol{\lambda}$ is high-dimensional. Thus, we propose to employ the GP-based Bayesian optimization[46–48] for solving Eq. (8), which can effectively tackle the above challenges. Start with Eq. (7) truncated to $\mathscr{N}$ terms, e.g., $\mathscr{N} = 3$ and correspondingly, $\boldsymbol{\lambda}$ is 12-dimensional: $(a_1, b_1, c_1, q_1, a_2, b_2, c_2, q_2, a_3, b_3, c_3, q_3)$. The GP-based Bayesian optimization takes the following steps: 1) Given some initial observations $\{\boldsymbol{\lambda}_k, \Pi(\boldsymbol{\lambda}_k) |\ k = 1, 2, \ldots, j_0\}$, a probabilistic model is built for Π using GP regression that has marginal closed-form for the *posterior* mean and variance. The number (j_0) and selection of initial data will be discussed later. 2) A cheap utility function, e.g., the lower confidence bound function[49] $\text{LCB}(\boldsymbol{\lambda})$ chosen in this paper, is optimized based on the *posterior* to decide where to take the next evaluation. 3) Augment the data with the next observation. These three steps are iterated until the objective function is less than preset tolerance ζ_{BO} or the number of iterations reaches the limit $\mathbb{I}_{\max}$. The GP has marginal closed-form for the *posterior* mean and variance, and hence is chosen for building probability measure over objective and acquisition functions. The uncertainty level determined from the GP is exploited to balance exploration against exploitation. Thus, the method is able to start with a small set of initial data and adaptively add more data as necessary at locations that can maximize information gain. The procedure of the GP-based Bayesian optimization employed in this work is outlined in Algorithm 1.

Algorithm 1 GP-based Bayesian optimization

Set ζ_{BO} and the searching space of $\boldsymbol{\lambda}$
for $i = 1, 2, \cdots, 10$ **do**
  Generate $\{\boldsymbol{\lambda}_k, \Pi(\boldsymbol{\lambda}_k) |\ k = 1, 2, \ldots, j_0\}$
  for $j = 0, 1, \ldots, \mathbb{I}_{\max}$ **do**
    Build a GP regression model for Π
    Optimize $\text{LCB}(\boldsymbol{\lambda})$ based on the mean and variance of GP regression to determine:

$$\boldsymbol{\lambda}_{j+1} = \underset{\boldsymbol{\lambda}}{\arg\min}\, \text{LCB}(\boldsymbol{\lambda})$$

    Evaluate $\Pi(\boldsymbol{\lambda}_{j+1})$
    if $\Pi(\boldsymbol{\lambda}_{j+1}) < \zeta_{BO}$ **then**
      End loop with $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_{j+1}$
    else
      Augment data set $\{\boldsymbol{\lambda}_k, \Pi(\boldsymbol{\lambda}_k) |\ k = 1, 2, \ldots, j_0 + j + 1\}$
    end if
  end for
end for
Output $\boldsymbol{\lambda}^*$ and $\Pi(\boldsymbol{\lambda}^*)$

To avoid the optimization falling into local minima and to ensure sufficient exploration of the parameter space, initial observations $\{\boldsymbol{\lambda}_k, \Pi(\boldsymbol{\lambda}_k) |\ k = 1, 2, \ldots, j_0\}$ should be sampled covering the range of searching space. In the meanwhile, the cost of constructing a GP model each time exhibits a cubic scaling with the number of observations. With more initial data, the cost of constructing

GP models and optimizing utility functions would dominate the total cost of Bayesian optimization and make the computation more expensive. Thus, in practice, less initial data (smaller j_0) is preferred in each Bayesian optimization loop. To ensure the optimization unbiased, we can perform Bayesian optimization for several times with different random initial observations, which is noted by the outer for-end loop in Algorithm 1. For instance, if 1000 initial random observations are considered sufficient to cover the searching space, we can divide the 1000 initial random observations into 10 groups each with 100 data and perform Bayesian optimization for 10 times using one group of initial data each time. The number of groups and the amount of data in each group are decided from the trade-off between two considerations: 1) using fewer initial observations can accelerate each optimization process; and 2) conducting more Bayesian optimization processes increases the costs because the objective function has to be evaluated more times.

Although theoretically sound, Bayesian optimization is practically limited to optimizing 10-20 parameters. This is due to the fact that the number of data samples required to cover the searching space exponentially increases with the dimension of the parameter space[50]. As a result, Bayesian optimization in high dimensional parameter space can be expensive. For a polymer solution system that entails a slow-decaying VACF(t) and hence a long memory, approximation of $K(t)$ for the entire $t \le t_{\text{cut}}$ typically requires to retain more than 5 terms in Eq. (7). The resulting $\boldsymbol{\lambda}$ consists of more than 20 parameters. Thus, directly applying Bayesian optimization to determine $K(t)$ for the entire $t \le t_{\text{cut}}$ is challenging. In the present work, we propose a two-stage Bayesian optimization method, as described in the following, to address this challenge.

First, we notice that $K(t)$ typically displays a fast-decaying short-time dynamics followed by a slow-decaying long tail. We specify a time scale t_{short} to divide $K(t)$ into the short-time and long-tail parts such that $|\text{VACF}(t)/\text{VACF}(0)| \le 10^{-1}$ for $t \le t_{\text{short}}$. A schematic diagram of different time scales is shown in Fig. 2. The

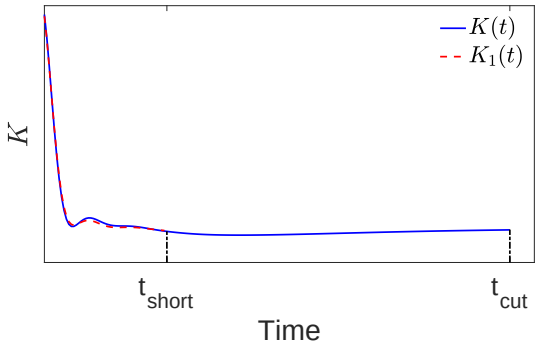

Fig. 2 Schematic diagram of different time scales in $K(t)$.

short-time $K(t \le t_{\text{short}})$, denoted as $K_1(t)$, due to its fast-decaying behavior, typically can be satisfactorily approximated by retaining 2-3 terms in Eq. (7); i.e., 8-12 parameters need to be optimized, which can be efficiently achieved using Bayesian optimization. Thus, in the *first* stage, Bayesian optimization is employed to solve Eq. (8) for $K_1(t)$ with $4\mathscr{N}_1$ (e.g., $\mathscr{N}_1 = 3$) parameters in $\boldsymbol{\lambda}$. To provide Bayesian optimization reasonable searching space for all parameters, we note $a_l, b_l, q_l > 0$, and c_l is related to a_l, b_l, q_l and must satisfy $|c_l| \le \frac{a_l b_l}{2q_l}$ (see §3.3 for explanation). The upper bounds for a_l and q_l can be as large as physically reasonable, e.g., in this work we gave $a_l \in [0, 60]$ and $q_l \in [0, 30]$. To specify the searching space for b_l, from Eq. (7) we know that b_l is directly related to $K(t=0)$. According to the relationship: $K(t=0) = \sum_{l=1}^{\mathscr{N}} b_l \approx \text{FACF}(t=0)$, we constrain $b_l \in [0, \text{FACF}(t=0)]$, where FACF is the force autocorrelation function and defined as FACF$=\langle F(t)F(0)\rangle$. The optimized parameters are denoted as $\boldsymbol{\lambda}_1^*$.

Second, approximating the entire $K(t)$ (up to t_{cut}) may require to supplement the $\mathscr{N}_1$ terms with more terms. Assume we need $\mathscr{N}_1 + \mathscr{N}_2$ (e.g., $3+3$) terms to accurately approximate the entire $K(t)$. The augmented $\mathscr{N}_2$ terms attempt to accurately approximate the slow-decaying long tail of $K(t)$ for $t_{\text{cut}} \ge t > t_{\text{short}}$. When we employ the Bayesian optimization to optimize $4(\mathscr{N}_1 + \mathscr{N}_2)$ parameters, although dimensionality of the parameter space is high, the optimized parameters $\boldsymbol{\lambda}_1^*$ obtained in the first stage can narrow down the searching space for the $4\mathscr{N}_1$ parameters, which can significantly reduce the number of iterations required in the optimization. Thus, in the *second* stage, Bayesian optimization is employed to solve Eq. (8) for the entire $K(t)$ ($t \le t_{cut}$) with $4(\mathscr{N}_1 + \mathscr{N}_2)$ parameters. The searching space for the $4\mathscr{N}_1$ parameters is constrained in $[(1-\alpha)\boldsymbol{\lambda}_1^*, (1+\alpha)\boldsymbol{\lambda}_1^*]$. Here, $\alpha < 1$ can start with a small value and gradually increase according to whether the preset tolerance can be reached within the maximum iterations allowed. In this work, $\alpha = 0.2$ was used in all cases. Given the searching space for the $4\mathscr{N}_1$ parameters and noting that $K(t=0) = \sum_{l=1}^{\mathscr{N}_1+\mathscr{N}_2} b_l \approx \text{FACF}(t=0)$, the upper limit of b_l in the $\mathscr{N}_2$ terms should be no greater than $\alpha \times \text{FACF}(t=0)$.

The procedure of the proposed two-stage Bayesian optimization is summarized in Algorithm 2. Although this approach requires two stages of Bayesian optimization, the number of iterations in each stage and total computational time can be significantly reduced, compared with the single-stage Bayesian optimization to achieve similar results. In this work, we set $\mathbb{I}_{\max} = 500$ and $\zeta_{BO} = 0.02$, and the Skopt package[47] was used to perform each Bayesian optimization.

Algorithm 2 Two-stage Bayesian optimization

Stage 1: Solve Eq. (8) for $t \le t_{\text{short}}$
Set $\mathscr{N}_1$ and the searching space: $a_l \in [0, 60]$, $b_l \in [0, \text{FACF}(t=0)]$, $c_l \in [-\frac{a_l b_l}{2q_l}, \frac{a_l b_l}{2q_l}]$, and $q_l \in [0, 30]$ for $l = 1, \cdots, \mathscr{N}_1$
Execute Algorithm 1
Output $\boldsymbol{\lambda}_1^*$ and $\Pi(\boldsymbol{\lambda}_1^*)$

Stage 2: Solve Eq. (8) for $t \le t_{\text{cut}}$
Set $\mathscr{N}_2$ to augment $\mathscr{N}_1$ s.t. $\mathscr{N} = \mathscr{N}_1 + \mathscr{N}_2$
Set the searching space for $\mathscr{N}_1$ terms: $[(1-\alpha)\boldsymbol{\lambda}_1^*, (1+\alpha)\boldsymbol{\lambda}_1^*]$, and the searching space for the rest $\mathscr{N}_2$ terms: $a_l \in [0, 60]$, $b_l \in [0, \alpha \times \text{FACF}(t=0)]$, $c_l \in [-\frac{a_l b_l}{2q_l}, \frac{a_l b_l}{2q_l}]$, $q_l \in [0, 30]$ for $l = 1, \cdots, \mathscr{N}_2$
Execute Algorithm 1
Output $\boldsymbol{\lambda}^*$ and $\Pi(\boldsymbol{\lambda}^*)$

3.2.2 Algebraic decay

In general, for fluids, including polymeric fluids (polymer solution or melt), the long-time behaviors of the VACF and memory kernel display hydrodynamic scaling, i.e., algebraic decay with the scaling $t^{-\frac{3}{2}}$ for $t \geq t_{\text{alg}}$ [35–38], where t_{alg} denotes the characteristic time of the emergence of algebraic decay. Different kinds of fluids have different t_{alg}. In the case of $t_{\text{alg}} < t_{\text{cut}}$, e.g., for a dilute polymer solution, the memory kernel for $t \geq t_{\text{alg}}$ simply follows the algebraic decay, and hence we only need to infer $K(t < t_{\text{alg}})$ via Bayesian optimization. In this section, we propose an active learning method to determine t_{alg} (emergence of the hydrodynamic algebraic decay) "on-the-fly" while generating the data of VACF in atomistic simulations.

To proceed, setting $X = g(t) = t^{-\frac{3}{2}}$ and $Y = \text{VACF}$ yields a linear regression model:

$$Y = \beta_1 X + \varepsilon \quad \text{for } t \geq t_{\text{alg}}\,, \tag{10}$$

where β_1 is the slope and ε is an identical independent Gaussian noise with zero mean and variance of σ_ε^2. Note since $\text{VACF} \to 0$ when $X \to 0$ ($t \to \infty$), the above linear regression model assumes a zero intercept. The task herein is to determine t_{alg} and β_1 via regression from the VACF data. The t_{alg} and β_1 sought can be used to predict the VACF from Eq. (10) beyond the dataset. Here, we require the standard deviation of the predicted VACF at t_{cut} is less than a preset tolerance, i.e., $\sigma(\text{VACF}(t_{\text{cut}})) \leq \zeta_{\text{AD}}$, where ζ_{AD} is the preset tolerance. (Recall t_{cut} is the time scale defined such that $|\text{VACF}(t)/\text{VACF}(0)| \leq 10^{-3}$ for $t > t_{\text{cut}}$.) We denote the latest time of the VACF dataset as t_{test}. For clarity, the different time scales are indicated in Fig. 3.

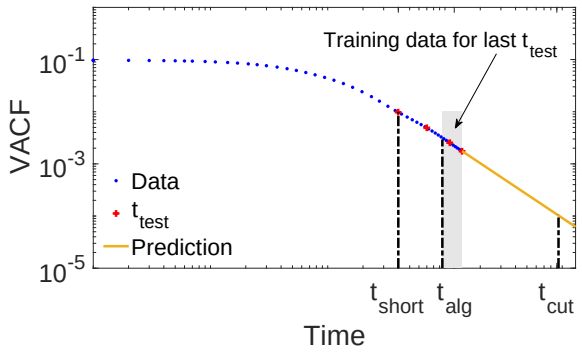

Fig. 3 Schematic diagram of different time scales in VACF(t).

The variance of the noise ε in Eq. (10) is first estimated from $\mathscr{G}_1$ training data by

$$\hat{\sigma}_\varepsilon^2 = \sum_{i=1}^{\mathscr{G}_1} (Y_i - \hat{Y}_i)^2/(\mathscr{G}_1 - 2) = \sum_{i=1}^{\mathscr{G}_1} (Y_i - \hat{\beta}_1 X_i)^2/(\mathscr{G}_1 - 2)\,, \tag{11}$$

where $\hat{\beta}_1$ can be estimated by

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{\mathscr{G}_1} X_i Y_i}{\sum_{i=1}^{\mathscr{G}_1} X_i^2}\,. \tag{12}$$

In Eqs. (11)-(12), the notation $\hat{\cdot}$ is used to denote the estimated values from the training data: $\{t_i, \text{VACF}(t_i)\}$ with $i = 1, 2, \ldots, \mathscr{G}_1$ and $t_i = t_{\text{test}} - (i-1)\Delta t$. Here, the size ($\mathscr{G}_1$) of dataset is determined from a theoretical argument: the standard deviation $\hat{\sigma}_\varepsilon$ calculated by Eq. (11) from $\mathscr{G}_1$ data has the desired probability of falling into the preset interval around the true standard deviation σ_ε [51,52]. Thereby, for $\hat{\sigma}_\varepsilon$ having 95% probability to fall into $[(1-0.15)\sigma_\varepsilon, (1+0.15)\sigma_\varepsilon]$, $\mathscr{G}_1 = 87$.

Given the estimated $\hat{\beta}_1$, we can forecast the VACF at any future time using Eq. (10) and determine t_{cut} such that $|\hat{\text{VACF}}(t)/\text{VACF}(0)| \leq 10^{-3}$ for $t > t_{\text{cut}}$. The standard deviation of $\text{VACF}(t_{\text{cut}})$ depends on: 1) how faithfully the data follow algebraic decay, i.e., the model in Eq. (10); and 2) whether there are sufficient training data for regression. Thus, by assuming that the VACF data follow algebraic decay and evaluating the standard deviation of $\text{VACF}(t_{\text{cut}})$, we first examine whether the training data are sufficient for desired regression accuracy. To this end, the standard deviation of $\text{VACF}(t_{\text{cut}})$ can be estimated from:

$$\sigma[\text{VACF}(t_{\text{cut}})] = \hat{\sigma}_\varepsilon \sqrt{\frac{1}{\mathscr{G}}\left[1 + \frac{(X_{\text{cut}} - \bar{X})^2}{Var(X)}\right]}\,, \tag{13}$$

where $\mathscr{G}$ is the number of training data; and $X_{\text{cut}} = t_{\text{cut}}{}^{-\frac{3}{2}}$. Here, $\bar{X}$ and $Var(X)$ are the mean and variance of X in the training data, respectively, which can be estimated via the Delta method [53] considering that the discrete t are equidistant, as: $\bar{X} \approx g(\bar{t}) = g(t_{\text{test}} - \frac{\mathscr{G}-1}{2}\Delta t)$ and $Var(X) \approx Var(t)[\frac{dg(\bar{t})}{dt}]^2 = \frac{1}{12}\Delta t^2(\mathscr{G}-1)(\mathscr{G}+1)[\frac{dg(\bar{t})}{dt}]^2$. Requiring $\sigma[\text{VACF}(t_{\text{cut}})] < \zeta_{\text{AD}}$, the number of training data ($\mathscr{G}$) required can be inversely solved from Eq. (13). In this work, the tolerance is set as $\zeta_{\text{AD}} = 10^{-4}$. Note that $\mathscr{G}$ must be no less than $\mathscr{G}_1$. The latter has been used to estimate $\hat{\sigma}_\varepsilon^2$ (the variance of the noise ε) in Eq. (11). Thus, $\mathscr{G} = \max\{\mathscr{G}, \mathscr{G}_1\}$.

After determining $\mathscr{G}$, we finally evaluate the standard deviation of $\text{VACF}(t_{\text{cut}})$ from the training data $\{t_i, \text{VACF}(t_i)\}$ with $t_i = t_{\text{test}} - (i-1)\Delta t$, $i = 1, 2, \ldots, \mathscr{G}$, as:

$$\sigma[\text{VACF}(t_{\text{cut}})] = \hat{\sigma}_{\varepsilon,\mathscr{G}} \sqrt{\frac{1}{\mathscr{G}} + \frac{(X_{\text{cut}} - \frac{1}{\mathscr{G}}\sum_{i=1}^{\mathscr{G}} X_i)^2}{\sum_{i=1}^{\mathscr{G}} (X_i - \bar{X})^2}}\,, \tag{14}$$

where $\hat{\sigma}_{\varepsilon,\mathscr{G}}$ is calculated from Eq. (11) from $\mathscr{G}$ training data. If such evaluated $\sigma[\text{VACF}(t_{\text{cut}})]$ satisfies $\sigma[\text{VACF}(t_{\text{cut}})] \leq \zeta_{AD}$, it indicates that the VACF data used for training follow algebraic decay. Hence, $t_{\text{alg}} = t_{\text{test}} - (\mathscr{G}-1)\Delta t$ characterizes the emergence of algebraic decay; and the VACF for $t > t_{\text{test}}$ can be predicted by Eq. (10). If $\sigma[\text{VACF}(t_{\text{cut}})] > \zeta_{AD}$, it implies that the VACF up to t_{test} has not displayed algebraic decay, and hence, more data of VACF for $t > t_{\text{test}}$ need to be generated and examined using the liner regression method. This process is iteratively repeated until $\sigma[\text{VACF}(t_{\text{cut}})] \leq \zeta_{\text{AD}}$ is satisfied. To determine the new t_{test} for the next iteration, we propose the following formula:

$$t_{\text{test}} = t_{\text{test}} + (1 - \frac{{\sigma_X}^2}{\hat{\sigma}_{\varepsilon,\mathscr{G}}^2})\mathscr{G}\Delta t, \tag{15}$$

where $\sigma_X = \zeta_{\text{AD}} / \sqrt{\frac{1}{\mathscr{G}} + \frac{(X_{\text{cut}} - \frac{1}{\mathscr{G}}\sum_{i=1}^{\mathscr{G}} X_i)^2}{\sum_{i=1}^{\mathscr{G}} (X_i - \bar{X})^2}}$. This formula assumes that the VACF data newly generated follow algebraic decay without noise, and hence provides the most conservative estimation of

the next t_{test}.

The proposed active learning method to determine t_{alg} in VACF is summarized in Algorithm 3.

Algorithm 3 Active learning of algebraic decay in VACF

Initialize $t_{\text{test}} = t_{\text{short}}$ and determine $\mathscr{G}_1$
do
  Estimate $\hat{\sigma}_\varepsilon$ by Eq. (11) from $\mathscr{G}_1$ training data
  Estimate $\hat{\beta}_1$ by Eq. (12) from $\mathscr{G}_1$ training data and determine t_{cut}
  Determine $\mathscr{G}$ from Eq. (13) by requiring $\sigma[\text{VACF}(t_{\text{cut}})] \leq \zeta_{\text{AD}}$ and take $\mathscr{G} = \max\{\mathscr{G}, \mathscr{G}_1\}$
  Compute $\hat{\beta}_1$ by Eq. (12) and $\sigma[\text{VACF}(t_{\text{cut}})]$ by Eq. (14) from $\mathscr{G}$ training data
  if $\sigma[\text{VACF}(t_{\text{cut}})] \leq \zeta_{\text{AD}}$ **then**
    End loop
  else
    Generate more VACF data until new t_{test} determined by Eq. (15)
  end if
while $\sigma[\text{VACF}(t_{\text{cut}})]) > \zeta_{AD}$
Output $t_{\text{alg}} = t_{\text{test}} - (\mathscr{G}-1)\Delta t$
Predict VACF$(t > t_{\text{test}})$ by Eq. (10)

If the determined $t_{\text{alg}} < t_{\text{cut}}$, the method to infer the memory kernel is revised as follows. $K(t \leq t_{\text{test}})$ is inferred via the two-stage Bayesian optimization described in §3.2.1. Note the inference herein is up to t_{test} instead of t_{alg}, considering the error in $K(t)$ (within ζ_{BO}) introduced by Bayesian optimization. The memory kernel after t_{alg} follows algebraic decay: $K(t \geq t_{\text{alg}}) = \beta_1' t^{-\frac{3}{2}}$. Using the data of $K(t_{\text{alg}} \leq t \leq t_{\text{test}})$ as training data, β_1' is determined via regression as: $\beta_1' = \frac{\sum_{i=1}^{\mathscr{G}'} t_i^{-\frac{3}{2}} K(t_i)}{\sum_{i=1}^{\mathscr{G}'} t_i^{-3}}$, where $t_i = t_{\text{alg}} + (i-1)\Delta t$, $i = 1, 2, \ldots, \mathscr{G}$. Such obtained $K(t)$ up to t_{cut} is then approximated by Eq. (7) truncated to $\mathscr{N}$ terms by fitting.

3.3 Extended dynamics

Approximating the memory kernel by a finite set of exponentially damped oscillators as in Eq. (7) would allow to replace the non-Markovian dynamic equation (Eq. (2)) with a Markovian dynamics extended in higher dimensions. By doing so, the expensive cost of solving the GLE can be significantly reduced, as has been evidenced in literature[23,28,54]. To this end, Eq. (7) is rewritten in a matrix form as:

$$K(t) = -\mathbf{A}_{ps} e^{-t\mathbf{A}_{ss}} \mathbf{A}_{sp} \,, \tag{16}$$

where $\mathbf{A}_{ps} = -\mathbf{A}_{sp}^T$. If we define the parameter matrix $\mathbf{A} = [0,\ \mathbf{A}_{ps};\ \mathbf{A}_{sp},\ \mathbf{A}_{ss}]$, it can be assembled from the parameters in Eq. (7) by:

$$\mathbf{A}_l = \left[\begin{array}{c|cc} 0 & \sqrt{\frac{b_l}{2} - \frac{q_l c_l}{a_l}} & \sqrt{\frac{b_l}{2} + \frac{q_l c_l}{a_l}} \\ \hline -\sqrt{\frac{b_l}{2} - \frac{q_l c_l}{a_l}} & a_l & \frac{1}{2}\sqrt{4q_l^2 + a_l^2} \\ -\sqrt{\frac{b_l}{2} + \frac{q_l c_l}{a_l}} & -\frac{1}{2}\sqrt{4q_l^2 + a_l^2} & 0 \end{array}\right] . \tag{17}$$

In Eq. (17), the top right block contributes to $\mathbf{A}_{ps}$; the bottom left contributes to $\mathbf{A}_{sp}$; and the 2×2 block on the bottom right constitutes $\mathbf{A}_{ss}$ that is a block diagonal matrix consisting of 2×2 blocks.

Given Eq. (16) and by introducing auxiliary variables $\mathbf{s}$, the extended Markovian dynamics is given by:

$$\begin{pmatrix} \dot{\mathbf{P}} \\ \dot{\mathbf{s}} \end{pmatrix} = \begin{pmatrix} \langle \mathbf{F} \rangle \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & \mathbf{A}_{ps} \\ \mathbf{A}_{sp} & \mathbf{A}_{ss} \end{pmatrix} \begin{pmatrix} M^{-1}\mathbf{P} \\ \mathbf{s} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{B}_s \end{pmatrix} \begin{pmatrix} 0 \\ \boldsymbol{\xi} \end{pmatrix} . \tag{18}$$

Here, $\boldsymbol{\xi}$ is a vector of uncorrelated Gaussian random variables with $\langle \boldsymbol{\xi}(t) \rangle = \mathbf{0}$ and $\langle \xi_{I,\mu}(t) \xi_{J,\nu}(0) \rangle = \delta_{IJ}\delta_{\mu\nu}\delta(t)$, where ξ_ν and ξ_μ denote the different elements of $\boldsymbol{\xi}$. To satisfy the second fluctuation-dissipation theorem[39], $\mathbf{B}_s \mathbf{B}_s^T = k_B T(\mathbf{A}_{ss} + \mathbf{A}_{ss}^T)$. We can write the parameter matrix $\mathbf{B} = \text{diag}(0, \mathbf{B}_s)$. To ensure $\mathbf{A}$ and $\mathbf{B}$ are both real number matrices, the parameters in Eq. (7) need to satisfy: $a_l \geq 0$, $b_l \geq 0$ and $|c_l| \leq \frac{a_l b_l}{2q_l}$.

Assuming the memory kernel can be approximated by Eq. (16), the extended dynamics in Eq. (18) is equivalent to the GLE in Eq. (2)[54] with the random force

$$\tilde{\mathbf{F}}(t) = -\int_0^t \mathbf{A}_{ps} e^{-(t-t')\mathbf{A}_{ss}} \mathbf{B}_s \boldsymbol{\xi}(t') dt' \,. \tag{19}$$

In the present work, the implicit velocity-Verlet temporal integrator[55] was used to numerically solve Eq. (18) in the CG simulations.

4 Results

The proposed CG modeling was assessed in a benchmark star-polymer solution system. In solutions, the structural and dynamic properties of star polymers depend on their concentrations. Therefore, to demonstrate the robustness of the proposed CG modeling approach, we considered the benchmark solution system at different concentrations. In the CG model, each star polymer is coarse-grained as a single CG particle; and the solvent DOFs are eliminated. The CG potential and memory kernel in the GLE (Eq. (2)) were constructed following the methodology described in §3. The dynamics of the CG system is governed by Eq. (18). We denote the CG model with the CG potential determined by the IBI method as "CG1" and the CG model using the DNN method to construct the CG potential as "CG2". We examined the accuracy and computational efficiency of the CG models in reproducing both structural and dynamic properties of the reference atomistic system. The structural properties include the RDF characterizing two-body correlations of star polymers, angular distribution function (ADF) characterizing three-body correlations, and also the averaged local Steinhardt parameters $\bar{q}_4$ and $\bar{q}_6$ that characterize the many-body correlations[56,57]. The CG potential determines the structural properties of the CG system. The dynamic properties include the VACF and diffusion coefficient as functions of time. The memory kernel along with the CG potential determine the dynamics of the CG system. The simulations were performed using LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator)[58].

4.1 Atomistic system

The atomistic system of star-polymer solution consists of Lennard-Jones(LJ) beads. More specifically, each star polymer consists of a core LJ bead and 10 identical arms with 3 LJ beads per arm, as illustrated in Fig. 4. The core LJ bead and the LJ beads in each arm are connected by finitely extensible nonlinear elastic (FENE) bonds. The solvent is also modeled by LJ beads identical to those in star polymers. Three concentrations were considered in this

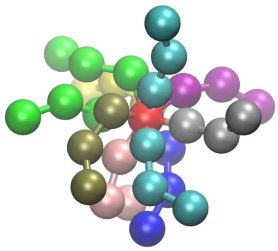

Fig. 4 Atomistic model of a star polymer consisting of 31 LJ beads: one core and 10 arms with 3 beads per arm.

work, as summarized in Table 1. The concentration is defined as the percentage-wise fraction of the LJ beads in star polymers vs. the total LJ beads in the polymer solution. The dynamics of the

Table 1 Atomistic systems with different concentrations.

System	Number of star polymers	Number of solvent beads	Concentration	Mole fraction
melt	2,500	0	100.0%	100.0%
dense	2,000	15,500	80.0%	11.4%
dilute	1,000	46,500	40.0%	2.1%

atomistic system is governed by the Hamiltonian:

$$H = \sum_{i=1}^{n} \frac{\mathbf{p}_i^2}{2m_i} + \sum_{i \neq j} E(r_{ij}) \,, \tag{20}$$

where $r_{ij} = |\mathbf{r}_{ij}| = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between two LJ beads; E denotes the total potential energy contributed by the inter-atomic and bonded potentials. The inter-atomic LJ potential adopts the purely repulsive Weeks-Chandler-Andersen (WCA) potential and is given by:

$$E_{WCA}(r) = \begin{cases} 4\varepsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6} + \frac{1}{4}] & r \leq r_c \\ \infty & r > r_c \end{cases} , \tag{21}$$

where $r_c = 2^{1/6}\sigma$ is the cutoff distance. The bonded interaction between connected LJ beads in star polymers is modeled as a spring with a FENE potential, i.e.,

$$E_{FENE}(r) = \begin{cases} -\frac{1}{2}kr_0^2 \ln[1 - (\frac{r}{r_0})^2] & r \leq r_0 \\ \infty & r > r_0 \end{cases} , \tag{22}$$

where $k = 30\varepsilon/\sigma^2$ is the spring constant, and $r_0 = 1.5\sigma$ is the maximum length of the FENE spring. In sum, we have:

$$E(\mathbf{r}_{ij}) = E_{WCA}(\mathbf{r}_{ij}) + E_{FENE}(\mathbf{r}_{ij}) \,. \tag{23}$$

The data of the atomistic systems were generated from MD simulations, which were performed in the canonical ensemble (NVT) using Noś e-Hoover thermostat with $k_BT = 1.0$ and the time step $\Delta t = 0.001\tau$. All the results in the present paper are expressed in the reduced LJ unit; i.e., the mass, length, energy, and time units are set as: $m = 1$, $\sigma = 1$, $\varepsilon = 1$, and $\tau = \sigma(m/\varepsilon)^{0.5} = 1$. To obtain accurate ensemble averages from the noisy data of MD simulations, 10 independent simulations with different random seeds were conducted for each system. In each simulation, the data after reaching thermal equilibrium were collected for computing the ensemble-averaged quantities of interest. The periodic cubic box of length 57.8647σ was used in all MD simulations. This size was chosen large enough such that the finite size effect on the VACF can be neglected.

4.2 Melt

We first considered a melt system of star polymers, where there is no solvent. Each star polymer is coarse-grained as a CG particle. The CG system consists of 2,500 CG particles. To construct the CG potential, the cutoff radius $R_{\text{cut}} = 6.4$ was employed in both IBI and DNN methods, which is at the first valley of the RDF. In the DNN method, one hidden layer with 10 nodes was used to build the DNN model of the CG potential. From the VACF data generated in MD simulation, we determined $t_{\text{cut}} = 18.5$, and $t_{\text{cut}} < t_{\text{alg}}$. Thus, we inferred the entire memory kernel $K(t)$ (up to t_{cut}) from the VACF data by employing the two-stage Bayesian optimization as described in §3.2.1 with $\mathscr{N}_1 = 3$, $\mathscr{N}_2 = 3$, and $t_{\text{short}} = 2.8$. Thereby, the inferred $K(t)$ was represented by $\mathscr{N} = \mathscr{N}_1 + \mathscr{N}_2 = 6$ terms of exponentially damped oscillators with the optimized parameters $\boldsymbol{\lambda}^*$. From $\boldsymbol{\lambda}^*$, the matrices $\mathbf{A}$ and $\mathbf{B}$ in §3.3 were assembled. With the extended dynamics fully determined, the CG simulation numerically solved Eq. (18) and computed the quantities that characterize the structural and dynamic properties of the CG system. The predictions by the CG simulation were compared with the MD simulation results.

4.2.1 Structural properties

The results on the structural properties are presented in Figs. 5-7. By comparison with the MD simulation results, we find that the CG model constructed can accurately reproduce all the structural properties. Since the structural properties of the CG system are determined by the CG potential, the results demonstrate the performance of the DNN and IBI methods employed for constructing the CG potential. From the results on the averaged local Steinhardt parameters $\bar{q}_4$ and $\bar{q}_6$, we find that the DNN method (in CG2) slightly outperforms the IBI method (in CG1) for reproducing the many-body correlations, which can be owing to the fact that the DNN method does not assume a pairwise potential but accounts for generally many-body contributions to the CG potential.

4.2.2 Dynamic properties

The memory kernel $K(t)$ inferred by employing the two-stage Bayesian optimization is plotted in Fig. 8. Since the CG potentials constructed by the IBI and DNN methods are not identical, the memory kernels in the two CG models are not the same.

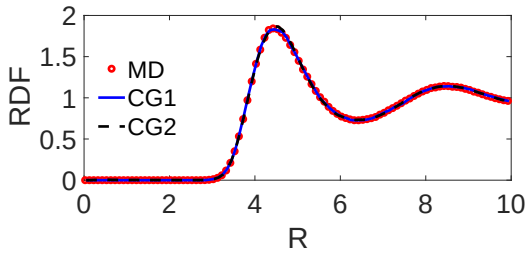

Fig. 5 RDF of star polymers in melt predicted by the CG models and compared with the MD simulation results.

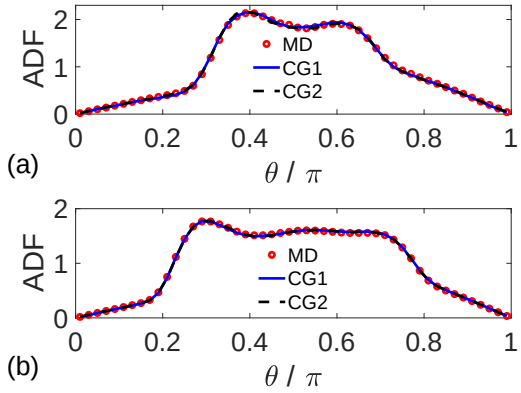

Fig. 6 ADF of star polymers in melt predicted by the CG models and compared with the MD simulation results. Here, the ADF was computed at two different positions: (a) $R = 4.5$ at the first peak of RDF and (b) $R = 6.4$ at the first valley of RDF.

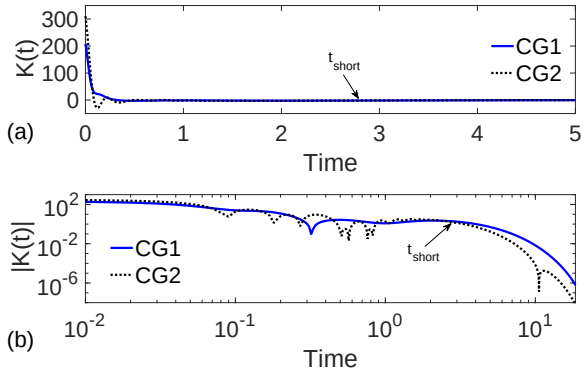

Fig. 8 Memory kernel inferred for star polymers in melt: (a) short-time $K(t)$ on linear scale and (b) long-time $|K(t)|$ on logarithmic scale. Here, $t_{\text{short}} = 2.8$.

The VACF and diffusion coefficient $D(t)$ predicted by the CG simulation beyond t_{cut} are depicted in Fig. 9. By comparison with the MD simulation results, only small discrepancy in the VACF's long tail is detected, which is less than $10^{-3} \times |\text{VACF}(0)|$. The overall good agreements demonstrate the accuracy of the constructed CG model in reproducing the dynamic properties of the reference atomistic system. In this regard, the CG models with the CG potential constructed by the IBI or DNN method do not show significant difference.

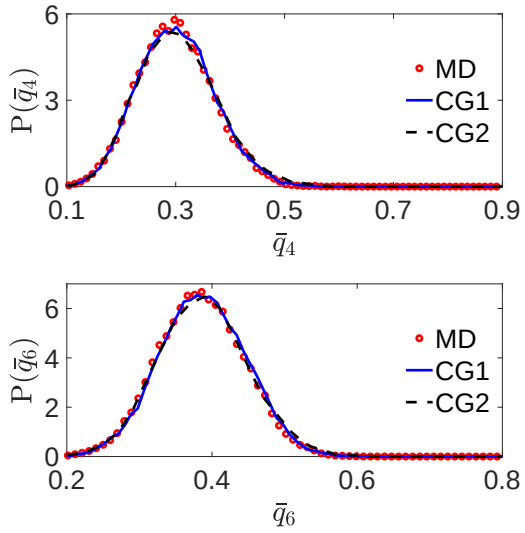

Fig. 7 Probability distributions of the averaged local Steinhardt parameters $\bar{q}_4$ and $\bar{q}_6$ of star polymers in melt predicted by the CG models and compared with the MD simulation results.

We note that Li et al.[23] also developed CG modeling for melt of star polymers. Two methods were compared in their work. One was based on the GLE and extended dynamics, similar to this work. However, the VACF and diffusion coefficient were not reproduced so accurately as in this work. The other method assumed pairwise interactions in the non-Markovian dynamics, which could reproduce the VACF and diffusion coefficient of the reference atomistic system with the same accuracy as in the present work. However, that method has two issues: 1) it requires to evaluate all pairwise dissipative and random interactions (within the cutoff) for each CG particle, which is more costly than directly evaluating the total dissipative and random forces on each CG particle; 2) its pairwise assumption cannot capture the many-body effect on the dynamics of polymers arising from solvent and hence is not applicable to polymers in solution.

4.3 Dense solution

We next studied a dense solution of star polymers. The atomistic system consists of 2,000 star polymers and 15,500 solvent beads. In CG modeling, each star polymer is coarse-grained as a CG particle; and the solvent DOFs are eliminated. Hence, the CG system consists of 2,000 CG particles. The cutoff radius $R_{\text{cut}} = 6.8$ (at the first valley of RDF) was employed in both methods for constructing the CG potential. In the DNN method, two hidden layers with 20 and 10 nodes, respectively (from the input end to the output end), were used to build the DNN model of the CG potential. From the VACF data generated in the MD simulation, we determined $t_{\text{cut}} = 10.2$, and $t_{\text{cut}} < t_{\text{alg}}$. Thus, we inferred the entire memory kernel $K(t)$ (up to t_{cut}) from the VACF data by employing the two-stage Bayesian optimization with $\mathcal{N}_1 = 3$, $\mathcal{N}_2 = 3$, and $t_{\text{short}} = 3.0$. Thereby, the inferred $K(t)$ was represented by $\mathcal{N} = \mathcal{N}_1 + \mathcal{N}_2 = 6$ terms of exponentially damped oscillators with the optimized parameters $\boldsymbol{\lambda}^*$. With the CG potential and memory kernel determined, the CG simulation solving the extended

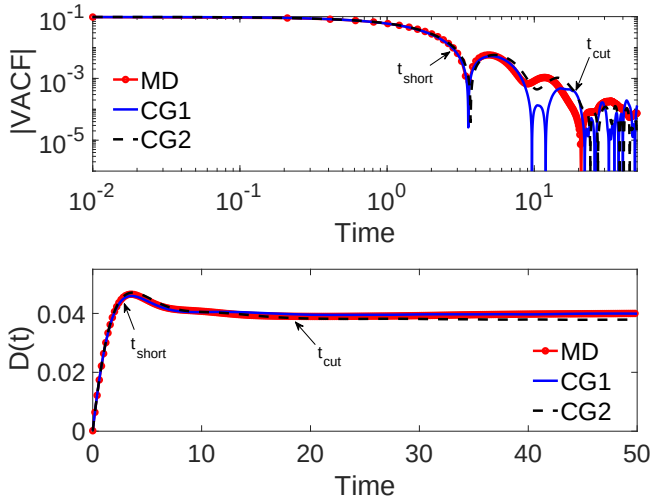

Fig. 9 Dynamic properties of star polymers in melt: the absolute value of VACF (on logarithmic scale) and diffusion coefficient $D(t)$ predicted by the CG models and compared with the MD simulation results. Here, $t_{\text{short}} = 2.8$ and $t_{\text{cut}} = 18.5$.

dynamics predicted the structural and dynamic properties of the CG system. The predictions by the CG simulation were compared with the MD simulation results.

4.3.1 Structural properties

The results by both CG and MD simulations on the structural properties are presented in Figs. 10-12. We find that the CG models constructed can accurately reproduce all the structural properties of the reference atomistic system. The performance of the IBI and DNN methods in correctly constructing the CG potential are comparable. The results on the averaged local Steinhardt parameters, especially, $\bar{q}_4$, indicate the DNN method outperforms the IBI method in reproducing the many-body correlations.

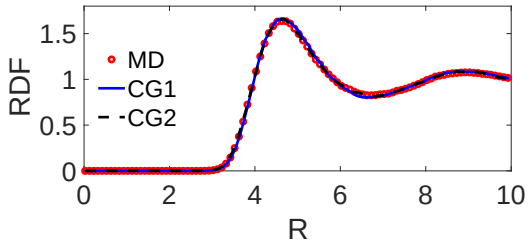

Fig. 10 RDF of star polymers in dense solution predicted by the CG models and compared with the MD simulation results.

4.3.2 Dynamic properties

The memory kernel $K(t)$ inferred by employing the two-stage Bayesian optimization is plotted in Fig. 13. The VACF and diffusion coefficient $D(t)$ predicted by the CG simulation (beyond t_{cut}) are depicted in Fig. 14. By comparison with the MD simulation results, only small discrepancy in the VACF's long tail (beyond t_{cut}) is noticed, which is less than $10^{-3} \times |\text{VACF}(0)|$. The overall good agreements demonstrate the accuracy of the constructed CG models in reproducing the dynamic properties of the reference atomistic system. In this regard, the CG models with the CG potential determined by the IBI or DNN method do not exhibit significant difference.

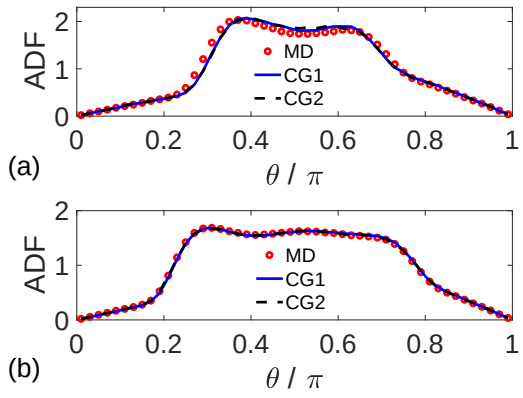

Fig. 11 ADF of star polymers in dense solution predicted by the CG models and compared with the MD simulation results. Here, the ADF was computed at two different positions: (a) $R = 4.7$ at the first peak of RDF and (b) $R = 6.8$ at the first valley of RDF.

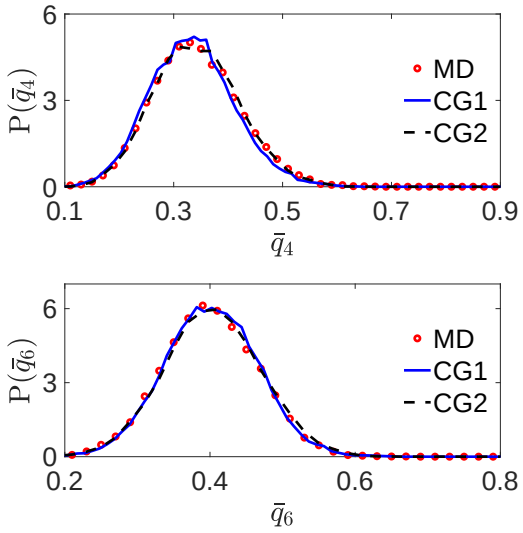

Fig. 12 Probability distributions of the averaged local Steinhardt parameters $\bar{q}_4$ and $\bar{q}_6$ of star polymers in dense solution predicted by the CG models and compared with the MD simulation results.

4.4 Dilute solution

Finally, we examined a dilute solution of star polymers. The atomistic system consists of 1,000 star polymers and 46,500 solvent beads. The CG system consists of 1,000 CG particles. The cutoff radius $R_{\text{cut}} = 7.4$ (at the first valley of RDF) was employed in both methods for constructing the CG potential. In the DNN method, 3 hidden layers with 40, 20 and 10 nodes, respectively (from the input end to the output end) were used to build the DNN model of the CG potential. Following the method described in §3.2.2, we determined $t_{\text{alg}} = 6.3$ and $t_{\text{cut}} = 93.7$ from the VACF data generated in the MD simulation. Since $t_{\text{alg}} < t_{\text{cut}}$, the memory kernel $K(t \le t_{\text{test}})$ was inferred using the two-stage Bayesian optimiza-

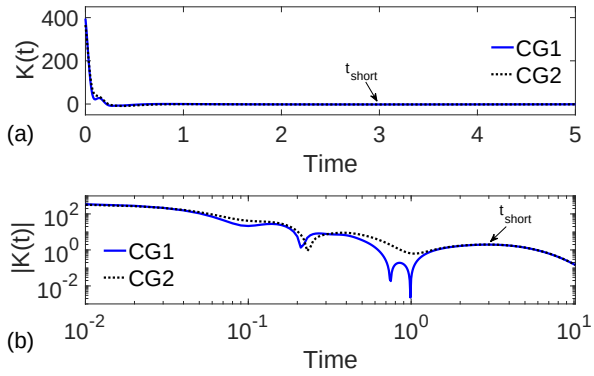

Fig. 13 Memory kernel inferred for star polymers in dense solution: (a) short-time $K(t)$ on linear scale and (b) long-time $|K(t)|$ on logarithmic scale. Here, $t_{\text{short}} = 3.0$.

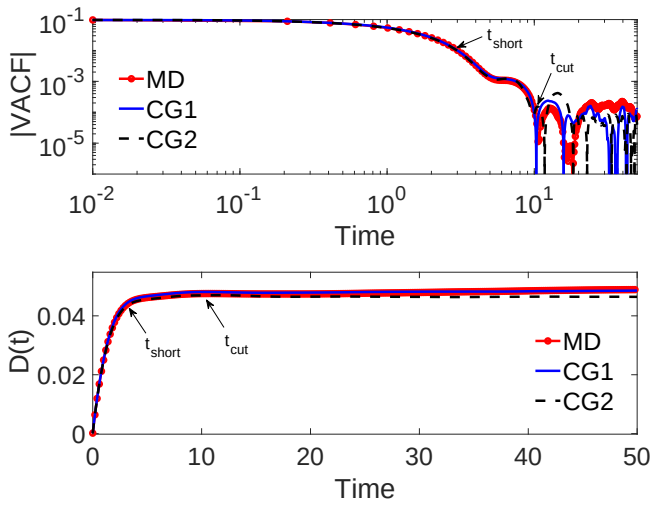

Fig. 14 Dynamic properties of star polymers in dense solution: the absolute value of VACF (on logarithmic scale) and diffusion coefficient $D(t)$ predicted by the CG models and compared with the MD simulation results. Here, $t_{\text{short}} = 3.0$ and $t_{\text{cut}} = 10.2$.

tion with $\mathcal{N}_1 = 3$, $\mathcal{N}_2 = 3$, $t_{\text{short}} = 3.8$, and $t_{\text{test}} = 7.2$. The memory kernel after t_{alg} followed algebraic decay: $K(t \geq t_{\text{alg}}) = \beta_1' t^{-\frac{3}{2}}$ with β_1' determined from the data of $K(t_{\text{alg}} \leq t \leq t_{\text{test}})$ via regression. The entire memory kernel $K(t \leq t_{\text{cut}})$ was then fitted by Eq. (7) truncated to $\mathcal{N} = 7$ terms in CG1 model and $\mathcal{N} = 8$ terms in CG2 model. The fitting parameters $\boldsymbol{\lambda}$ were then used to assemble the matrices **A** and **B** for the extended dynamics in §3.3. The CG simulations solving the extended dynamics predicted the structural and dynamic properties of the CG system. The predictions by the CG simulations were compared with the MD simulation results.

4.4.1 Structural properties

The results by both CG and MD simulations on the structural properties are presented in Figs. 15-17. The CG models constructed can accurately reproduce all the structural properties of the reference atomistic system. The overall performance of the IBI and DNN methods in correctly constructing the CG potential are comparable. However, the DNN method (in CG2) is superior to the IBI method (in CG1) in reproducing the many-body correlations, which is indicated by the results on the averaged local Steinhardt parameters $\bar{q}_4$ and $\bar{q}_6$ in Fig. 17.

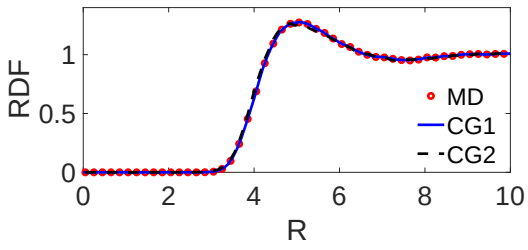

Fig. 15 RDF of star polymers in dilute solution predicted by the CG models and compared with the MD simulation results.

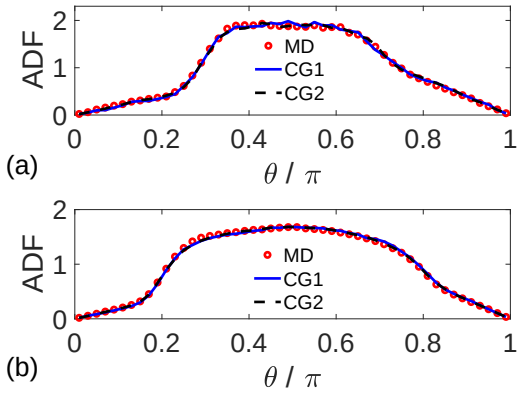

Fig. 16 ADF of star polymers in dilute solution predicted by the CG models and compared with the MD simulation results. Here, the ADF was computed at two different positions: (a) $R = 5.2$ at the first peak of RDF and (b) $R = 7.4$ at the first valley of RDF.

4.4.2 Dynamic properties

The VACF of star polymers in dilute solution displayed a very slowly decaying VACF, see Fig. 18. The finite-size effect of periodic box in MD simulations can cause spurious oscillations in the long tail of slow-decaying VACF, which in turn may affect accurately detecting hydrodynamic scaling behavior (algebraic decay with time) of VACF. Thus, we enlarged the periodic cubic box used in the MD simulation by two times but kept the same concentration of star polymers. The resulting length of periodic box is 115.7295σ, which was filled with 8,000 star polymers and 372,000 solvent beads. From the VACF data of this larger atomistic system, employing the method described in §3.2.2 resulted in the same t_{alg}, which confirmed the accuracy of the determined t_{alg}. The comparison of the VACF is depicted in Fig. 18, which also indicates that the long-time VACF (of the larger atomistic system) without spurious oscillations consistently follow the theoretical prediction of algebraic decay.

The memory kernel $K(t)$ inferred using the method described in §3.2.2 is presented in Fig. 19. The VACF and diffusion coefficient $D(t)$ predicted by the CG simulations are shown in Fig. 20. Those results are compared with the MD simulation results for $t \leq t_{\text{alg}}$ and compared with the theoretical prediction of algebraic decay

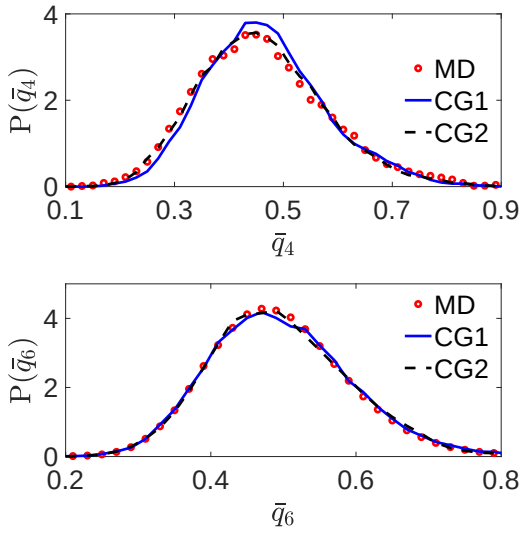

Fig. 17 Probability distributions of the averaged local Steinhardt parameters $\bar{q}_4$ and $\bar{q}_6$ of star polymers in dilute solution predicted by the CG models and compared with the MD simulation results.

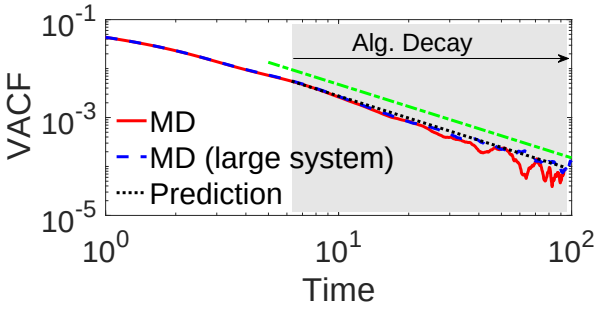

Fig. 18 Comparison of VACFs of the original atomistic system and a two-times larger system of star polymers in dilute solution. The slope $-\frac{3}{2}$ (green dash dot line) is drawn for reference.

for $t > t_{\text{alg}}$. Good agreements are achieved for either CG model (with the CG potential constructed by the IBI or DNN method).

4.5 Computational efficiency of CG simulations

In this section, we assess the cost of CG simulations vs. the reference atomistic simulations. For comparison, each simulation was conducted for $10^3\tau$ via serial computing on Intel Core i5-6500 CPU @ 3.20GHz. The computer time spent in each simulation was summarized in Table 2. Note the computer time reported herein for CG simulations does not include the time used to construct the CG models. Owing to the larger time step permitted and fewer total DOFs in CG simulations, we anticipate the CG simulations are more efficient than the reference atomistic simulations. Hence, in Table 2, we further evaluated the speedup factors of CG simulations.

5 Conclusion

We have presented a data-driven CG modeling approach for polymers in solution, which can reproduce the dynamic as well as structural properties of the reference atomistic system. The CG

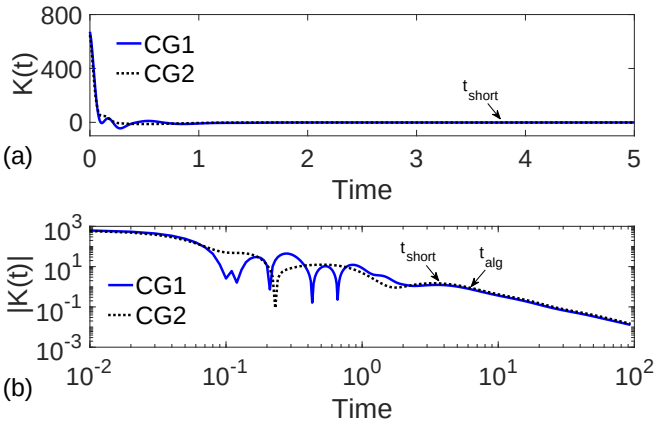

Fig. 19 Memory kernel inferred for star polymers in dilute solution: (a) short-time $K(t)$ on linear scale and (b) long-time $|K(t)|$ on logarithmic scale. Here, $t_{\text{short}} = 3.8$ and $t_{\text{alg}} = 6.3$.

Table 2 Comparison of computational cost of CG simulations vs. reference MD simulations.

System	Simulation	Δt (τ)	Cost (s)	Speedup Factor
melt	MD	0.001	6271	-
	CG1	0.1	59	106.3
	CG2	0.1	181	34.6
dense solution	MD	0.001	6117	-
	CG1	0.1	47	130.1
	CG2	0.1	175	35.0
dilute solution	MD	0.001	5485	-
	CG1	0.1	29	189.1
	CG2	0.1	88	62.3

modeling is built upon the framework of GLE. The non-Markovian memory kernel is inferred from the atomistic simulation data of VACF via two-stage GP-based Bayesian optimization. The uncertainty level determined from the GP enables the optimization to balance exploration against exploitation, leading to adaptive sampling of data with information gain maximized. Considering that the long-time behaviors of VACF and memory kernel for polymer solutions can exhibit hydrodynamic scaling (algebraic decay with time), we have proposed an active learning method to "on-the-fly" determine the emergence of hydrodynamic scaling while the atomistic simulations are generating the data of VACF. In addition, we have compared the DNN and IBI methods for constructing the CG potential. With the memory kernel and CG potential determined, the GLE is mapped onto an extended Markovian process to circumvent the expensive cost of directly solving the GLE. We have assessed the accuracy and computational efficiency of the proposed CG modeling in a benchmark polymer solution system at three representative concentrations.

By comparing with the atomistic simulation results, we have demonstrated the proposed CG modeling can robustly and accurately reproduce both the structural and dynamic properties of polymers, regardless of the solution concentrations. Moreover, the CG simulations have fewer total DOFs, permit larger time steps, and thereby are much more efficient than the reference atomistic simulations. Employing the DNN or IBI method for constructing the CG potential does not affect the effectiveness of the

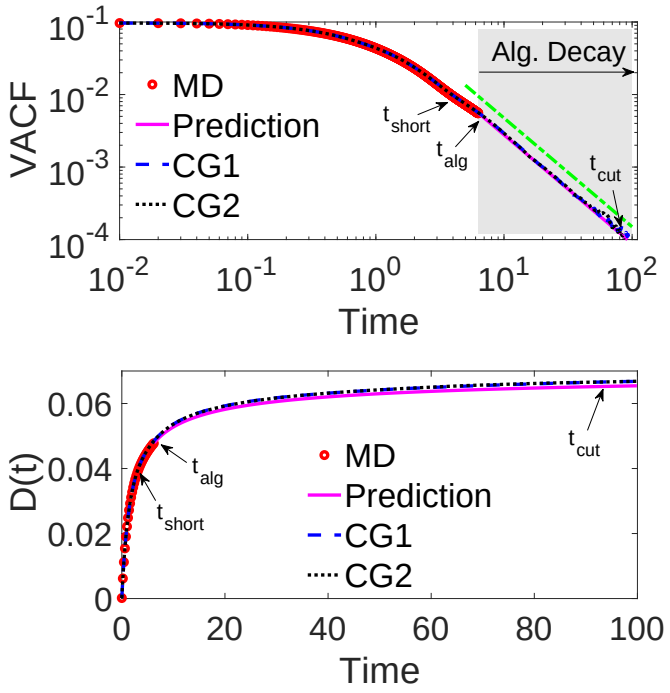

Fig. 20 Dynamic properties of star polymers in dilute solution: the VACF (on logarithmic scale) and diffusion coefficient $D(t)$ predicted by the CG models and compared with the MD simulation results for $t \leq t_{\mathsf{alg}}$ and theoretical prediction of algebraic decay for $t > t_{\mathsf{alg}}$. Here, $t_{\mathsf{short}} = 3.8$, $t_{\mathsf{alg}} = 6.3$ and $t_{\mathsf{cut}} = 93.7$.

proposed two-stage Bayesian optimization for inference of memory kernel and reproducing the dynamic properties. However, we have found that the DNN method outperforms the IBI method in reproducing many-body structural correlations characterized by the averaged local Steinhardt parameters. As the concentration of polymer solution decreases, the solvent-mediated many-body effect becomes more pronounced, and hence the CG potential becomes more complex and nonlinear, which is reflected by more layers and nodes required in the DNN representation of the CG potential. In spite of superior accuracy, the CG simulation with the DNN model of CG potential is significantly slower than the CG simulation with the IBI-determined potential. The dominant cost of calling the DNN model in CG simulations stems from assembly of the descriptor matrix $\mathbf{D}_I$ as the input of DNN. Future efforts aiming to reduce this cost would be worthwhile. Another avenue for future work would be to explore how the DNN's representation of the many-body potential can sensibly be decomposed into a sum of two-body, three-body, four-body, ... contributions. This effort may shed a light on how important the many-body interactions are, and how they can be represented such that potentially more efficient CG simulation schemes could be constructed.

Although a model system of star polymers was chosen to assess the accuracy and computational efficiency of the proposed CG modeling, the two-stage GP-based Bayesian optimization for inference of memory kernel and the active learning method to determine the emergence of hydrodynamic scaling in VACF are anticipated to be applicable to general polymer solution systems. We note that there may be cases that require more than 10 terms truncated in Eq. (7) to approximate the entire memory kernel, for which more than 40 parameters need to be optimized in the second stage of Bayesian optimization. For those cases, the second stage of Bayesian optimization can be expensive since the cost increases exponentially with the dimension of parameters. To tackle this issue, we could potentially employ the feature space-based Bayesian optimization[59–61], which can effectively reduce the dimensionality of the optimization problem by embedding the high dimensional parameters $\boldsymbol{\lambda} \in \mathbb{R}^{4\mathcal{N}}$ onto a low-dimensional feature space $\boldsymbol{\lambda}' \in \mathbb{R}^d$, where $d \ll 4\mathcal{N}$. By jointly learning *i*) the feature mapping: $\boldsymbol{\lambda} \to \boldsymbol{\lambda}'$, *ii*) the reconstruction mapping: $\boldsymbol{\lambda}' \to \boldsymbol{\lambda}$, and *iii*) GP model for $\Pi(\boldsymbol{\lambda}')$, the acquisition function $\mathrm{LCB}(\boldsymbol{\lambda}')$ can be optimized in the low-dimensional feature space, thereby effectively reducing the cost of Bayesian optimization for high-dimensional cases. The study in this regard will be in our future work.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1761068. The authors also thank the two anonymous reviewers for their insightful comments and suggestions that helped improve the manuscript.

References

1 M. G. Saunders and G. A. Voth, *Annual review of biophysics*, 2013, **42**, 73–93.
2 S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid and A. Kolinski, *Chemical reviews*, 2016, **116**, 7898–7936.
3 M. Dinpajooh and M. G. Guenza, *Soft matter*, 2018, **14**, 7126–7144.
4 K. M. Salerno, A. Agrawal, D. Perahia and G. S. Grest, *Physical review letters*, 2016, **116**, 058302.
5 A. Gooneie, S. Schuschnigg and C. Holzer, *Polymers*, 2017, **9**, 16.
6 J. Kleinjung and F. Fraternali, *Current opinion in structural biology*, 2014, **25**, 126–134.
7 T. T. Pham, M. Bajaj and J. R. Prakash, *Soft Matter*, 2008, **4**, 1196–1207.
8 R. Chudoba, J. Heyda and J. Dzubiella, *Soft matter*, 2018, **14**, 9631–9642.
9 G. Sevink and J. Fraaije, *Soft Matter*, 2014, **10**, 5129–5146.
10 Z. G. Mills, W. Mao and A. Alexeev, *Trends in biotechnology*, 2013, **31**, 426–434.
11 J. Mu, R. Motokawa, C. D. Williams, K. Akutsu, S. Nishitsuji and A. J. Masters, *The Journal of Physical Chemistry B*, 2016, **120**, 5183–5193.
12 L. Rovigatti, N. Gnan, L. Tavagnacco, A. J. Moreno and E. Zaccarelli, *Soft matter*, 2019, **15**, 1108–1119.
13 P. Bełdowski, P. Weber, A. Dėdinaitė, P. M. Claesson and A. Gadomski, *Soft matter*, 2018, **14**, 8997–9004.
14 D. Reith, M. Pütz and F. Müller-Plathe, *Journal of computational chemistry*, 2003, **24**, 1624–1636.

15 A. P. Lyubartsev and A. Laaksonen, *Physical Review E*, 1995, **52**, 3730.

16 S. Izvekov and G. A. Voth, *The Journal of Physical Chemistry B*, 2005, **109**, 2469–2473.

17 M. S. Shell, *The Journal of chemical physics*, 2008, **129**, 144108.

18 T. Sanyal and M. S. Shell, *The Journal of chemical physics*, 2016, **145**, 034109.

19 S. T. John and G. Csányi, *The Journal of Physical Chemistry B*, 2017, **121**, 10934–10949.

20 L. Zhang, J. Han, H. Wang, R. Car and W. E, *The Journal of chemical physics*, 2018, **149**, 034101.

21 J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. de Fabritiis, F. Noé and C. Clementi, *ACS Central Science*, 2019, **5**, 755–767.

22 A. Davtyan, J. F. Dama, G. A. Voth and H. C. Andersen, *The Journal of Chemical Physics*, 2015, **142**, 154104.

23 Z. Li, H. S. Lee, E. Darve and G. E. Karniadakis, *The Journal of chemical physics*, 2017, **146**, 014104.

24 L. Ma, X. Li and C. Liu, *The Journal of chemical physics*, 2016, **145**, 204117.

25 Y. Yoshimoto, Z. Li, I. Kinefuchi and G. E. Karniadakis, *The Journal of chemical physics*, 2017, **147**, 244110.

26 G. Jung, M. Hanke and F. Schmid, *Journal of chemical theory and computation*, 2017, **13**, 2481–2488.

27 H. S. Lee, S.-H. Ahn and E. F. Darve, *The Journal of chemical physics*, 2019, **150**, 174113.

28 S. Wang, Z. Li and W. Pan, *Soft Matter*, 2019, **15**, 7567–7582.

29 X. Bian, C. Kim and G. E. Karniadakis, *Soft Matter*, 2016, **12**, 6331–6346.

30 H. Lei, N. A. Baker and X. Li, *Proceedings of the National Academy of Sciences*, 2016, **113**, 14183–14188.

31 I. Y. Lyubimov, J. McCarty, A. Clark and M. G. Guenza, *The Journal of Chemical Physics*, 2010, **132**, 224903.

32 I. Lyubimov and M. G. Guenza, *Phys. Rev. E*, 2011, **84**, 031801.

33 L. Ma, X. Li and C. Liu, *Journal of Computational Physics*, 2019, **380**, 170–190.

34 G. Jung, M. Hanke and F. Schmid, *Soft matter*, 2018, **14**, 9368–9382.

35 A. McDonough, S. Russo and I. Snook, *Physical Review E*, 2001, **63**, 026109.

36 D. Lesnicki, R. Vuilleumier, A. Carof and B. Rotenberg, *Physical review letters*, 2016, **116**, 147804.

37 R. Dib, F. Ould-Kaddour and D. Levesque, *Physical Review E*, 2006, **74**, 011202.

38 N. Corngold, *Physical Review A*, 1972, **6**, 1570.

39 R. Kubo, *Rep. Prog. Phys.*, 1966, **29**, 255.

40 H. Wang, L. Zhang, J. Han and E. Weinan, *Computer Physics Communications*, 2018, **228**, 178–184.

41 J. Heaton, *Introduction to neural networks with Java*, Heaton Research, Inc., 2008.

42 H. B. Demuth, M. H. Beale, O. De Jess and M. T. Hagan, *Neural network design*, Martin Hagan, 2014.

43 V. Agrawal, G. Arya and J. Oswald, *Macromolecules*, 2014, **47**, 3378–3389.

44 V. Ruhle, C. Junghans, A. Lukyanov, K. Kremer and D. Andrienko, *Journal of Chemical Theory and Computation*, 2009, **5**, 3211–3223.

45 R. Zwanzig, *Annual Review of Physical Chemistry*, 1965, **16**, 67–102.

46 J. Snoek, H. Larochelle and R. P. Adams, *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., 2012, pp. 2951–2959.

47 G. Louppe and M. Kumar, *Bayesian optimization with skopt*, `https://scikit-optimize.github.io/notebooks/bayesian-optimization.html`, 2016.

48 GPyOpt, *Machine Learning Group - University of Sheffield*, `https://sheffieldml.github.io/GPyOpt/`, 2018.

49 J. Snoek, H. Larochelle and R. P. Adams, *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., 2012, pp. 2951–2959.

50 B. Shahriari, K. Swersky, Z. Wang, R. P. Adams and N. de Freitas, *Proceedings of the IEEE*, 2016, **104**, 148–175.

51 J. A. Greenwood and M. M. Sandomire, *Journal of the American Statistical Association*, 1950, **45**, 257–260.

52 M. Desu and D. Raghavarao, *Sample Size Methodology*, Academic Press, New York, 1990.

53 T. L. Kelley, *The Elementary School Journal*, 1928, **29**, 147–149.

54 M. Ceriotti, G. Bussi and M. Parrinello, *Journal of Chemical Theory and Computation*, 2010, **6**, 1170–1180.

55 A. Brünger, C. L. Brooks III and M. Karplus, *Chemical physics letters*, 1984, **105**, 495–500.

56 P. J. Steinhardt, D. R. Nelson and M. Ronchetti, *Physical Review B*, 1983, **28**, 784.

57 W. Lechner and C. Dellago, *The Journal of chemical physics*, 2008, **129**, 114707.

58 S. Plimpton, *Journal of Computational Physics*, 1995, **117**, 1 – 19.

59 Z. Wang, M. Zoghi, F. Hutter, D. Matheson and N. De Freitas, Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013, p. 1778–1784.

60 R. Garnett, M. A. Osborne and P. Hennig, Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, Arlington, Virginia, USA, 2014, p. 230–239.

61 R. Moriconi, M. P. Deisenroth and K. S. S. Kumar, *High-dimensional Bayesian optimization using low-dimensional feature spaces*, 2019.