Identifying polymer states by machine learning

Qianshi Wei,[1] Roger G. Melko,[1,2] and Jeff Z. Y. Chen[1,*]
[1]*Department of Physics and Astronomy, University of Waterloo, Waterloo N2L 3G1, Canada*
[2]*Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada*
(Received 5 January 2017; published 30 March 2017)

The ability of a feed-forward neural network to learn and classify different states of polymer configurations is systematically explored. Performing numerical experiments, we find that a simple network model can, after adequate training, recognize multiple structures, including gaslike coil, liquidlike globular, and crystalline anti-Mackay and Mackay structures. The network can be trained to identify the transition points between various states, which compare well with those identified by independent specific-heat calculations. Our study demonstrates that neural networks provide an unconventional tool to study the phase transitions in polymeric systems.

DOI: 10.1103/PhysRevE.95.032504

I. INTRODUCTION

Machine learning is an active research branch of computer science, which has vastly matured in recent years [1–6]. Artificial neural networks (NN) are a computational implementation of machine learning that have demonstrated surprising capability in recognizing patterns of enormous complexity, after appropriately trained by human or self-trained through learning mechanisms [7–13]. Originally motivated by the desire to establish an algorithmic model of the neuronal configuration of a mammalian brain, one finds that an artificial NN with a very simple underlying structure can successfully perform many complex tasks. For example, simple NN models have shown transformative success in handwriting and speech recognition [14–17].

The application of machine learning in computer simulations of molecular systems has only recently emerged. Machine learning has been proposed as a method for obtaining approximate solutions for the equations of density-functional theory [18], for constructing force fields for molecular dynamics [19], for identifying three-dimensional structures in polymorphic systems [20], and for providing an inexpensive impurity solver for dynamical mean-field theory [21]. More recently, a simple supervised NN was implemented to classify phases in a two-dimensional (2D) spin model and to identify critical temperatures between various phases directly from configurations produced by Monte Carlo (MC) simulations [22]. This comes on the crest of a wave of work exploring the hybridization of MC and machine learning [23–26] and the deeper connections between techniques such as deep learning to theoretical concepts familiar from statistical mechanics, such as the renormalization group [27,28].

Here we demonstrate the capability of a standard NN model in recognizing various configurations produced from MC simulations of polymer models, identifying disordered, partially ordered, and ordered states such as coil, liquidlike globular, and two low-energy crystalline structures. The physical system we discuss has a long history and hence is used as a typical example of a classical system displaying phase transitions that can be conveniently studied by using a hybridization of conventional simulation and machine-learning techniques.

*jeffchen@uwaterloo.ca

II. THE FEED-FORWARD NEURAL NETWORK

A standard feed-forward NN for supervised learning consists of three layers of neurons or "nodes," input (i), hidden (h), and output (o), each containing N_i, N_h, N_o neurons (see Fig. 1). These nodes are connected through edges (representing model weights), forming a fully connected graph between (but not among) each neuron layer. In a typical image recognition application, a 2D picture is discretized into N_i pixels and the values of pixel's intensity are then fed into the input layer for either training or testing purposes. The configurations of a 2D Ising model, where N_i spins have binary values, share some similarity to the standard image recognition problem, as each configuration is a "snapshot" of a 2D pattern of pixels. However, in a simple feed-forward NN, the details of this dimension and locality are lost as pixels are simply sent into input nodes as a one-dimensional vector [22].

A simulated, 3D polymer configuration is mathematically represented by its $3N$ spatial coordinates of the N connected monomers. In order to proceed, one must decide how this information is fed into the NN. One option is to naively adopt the image recognition idea above by digitizing the 3D space into a grid and assigning a value of "occupied" or "unoccupied" to a particular grid point, describing the occupancy of monomers. The entire pattern would then be used for analysis. Alternatively, here we analyze a polymer configuration by directly feeding $3N$ coordinates into $N_i = 3N$ input nodes. Then, to implement supervised learning, the NN is trained to analyze the relationship between the monomer position vector and a configurational pattern corresponding to the "label" in the output layer. In total $N_h = 100$ neurons are used and the number of the output nodes, N_o, is set to 2 or 3 depending on the physical problems described below. As a technical note, during the training session, we adopted cross entropy as the cost function and used 50% dropout rate in determination of the parameters related to the hidden layer, which is a regularization method to avoid overfitting [29].

III. IDENTIFYING POLYMER STRUCTURES

A. Polymer states

Two polymer models are used here, both consisting of $N = 102$ connected monomers interacting through a typical short-range hard-core repulsion and a slightly longer-range

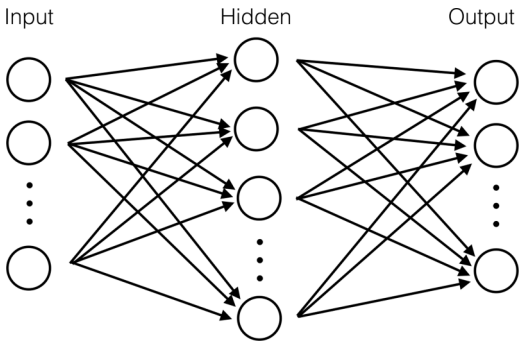

FIG. 1. Sketch of a fully connected feed-forward artificial neural network. The circles represent neuron nodes connected layer by layer via various weight and activation parameters.

attraction. One of the polymer models is known to exhibit four different states, illustrated in Fig. 2, separated by phase transitions of finite-size characteristics [30–32]. Within the first model, the bonded monomers interact with each other by a spring potential, identical to the one used in the Gaussian model (GSM) with a Kuhn length a [33]. Two nonbonded monomers interact with each other by a square-well potential: Below a square distance of $0.81a^2$ the monomers repel through the excluded-volume interaction, and between $0.81a^2$ and $2a^2$ the monomers experience an attraction of magnitude $-\epsilon$. Within the second model, the bonded monomers are modeled by a particular implementation of the finitely extensible nonlinear elastic (FENE) model, and the monomers interact with each other in the form of the Lennard-Jones (LJ) potential with a potential well depth of $-\epsilon$. The selection of FENE and LJ functions exactly matches those used in Ref. [32]. In this paper we refer to the first model as GSM and to the second as FENE for simplicity. More details are defined in Appendixes A and B.

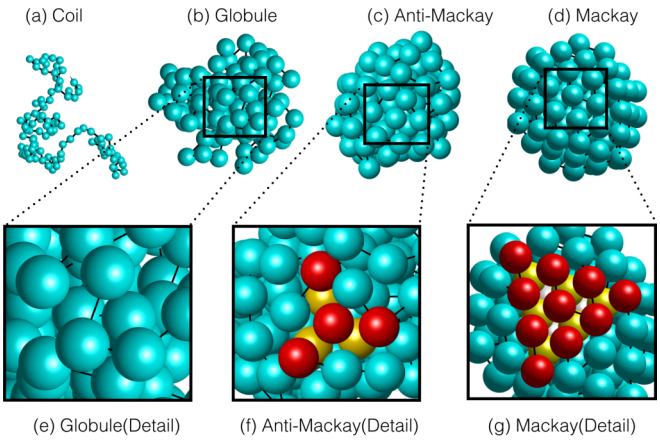

FIG. 2. Typical configurations of a polymer in (a) coil, (b) globular, (c) anti-Mackay, and (d) Mackay states as the temperature is lowered. The $N = 102$ monomers are represented by blue spheres, except for those in the last two plots. At the liquidlike globular state, the monomer positions are disordered, shown in (e). Both anti-Mackay and Mackay states are crystalline and differ from each other by the monomer stacking symmetries, demonstrated by the red (dark gray) and yellow (light gray) spheres in (f) and (g).

B. Coil-to-globule transition

Monte Carlo simulations which incorporate the Boltzmann weight were used for GSM to produce 5×10^3 independent configurations at every specified temperature k_BT/ϵ where k_B is the Boltzmann constant. Assessing the data shown in Fig. 3(a) for both reduced mean–square radius of gyration, $S^2 \equiv \langle R_g^2 \rangle / a^2$, and mean-square deviation of the total energy from its average (which is proportional to the specific heat), $\tilde{C} \equiv [\langle E_{\rm int}^2 \rangle - \langle E_{\rm int} \rangle^2]/N^2\epsilon^2$, we observe that a coil- (C) to-globule (G) phase transition takes place at $k_BT/\epsilon \approx 2.0$, corresponding to the location of the peak in $\tilde{C}$.

To demonstrate the capability of NN in recognizing C and G, we performed three numerical experiments. First, 3×10^3 polymer configurations (specified by the coordinates of monomers after setting $a = 1$) at every specified temperature within the ranges [0.5,1.5] and [2.5,3.5] were selected as training sets for the G and C states, respectively. The two normalized output neurons were designated as the G and C labels, which during training were assigned to have values $\nu = 1$ for the corresponding state and $\nu = 0$ otherwise. No other information or estimators typical of MC simulations, such as S^2 or $\tilde{C}$, were used in the training. Once the NN was adequately trained and all NN parameters were fixed, we input 500 new configurations at every temperature in the range [0.5,5.5], which were not used in the training session, as the testing set. The averaged test values of the two output neurons, ν, are plotted in Fig. 3(b), forming two curves behind the filled and open squares. These curves cross each other, identifying a C-to-G transition at $k_BT/\epsilon = 2.03$, in agreement with the location of the $\tilde{C}$ peak, regardless of the fact that the NN model was trained in temperature ranges farther away from the transition point.

The C and G states have distinguishably different dimensions, represented by S^2. To show that NN is not simply recognizing C and G from their different overall sizes, in our second experiment we normalized all coordinates of the training and testing sets by a factor $1/S$. On average, the polymer configurations now have the same normalized radius of gyration (=1) across the entire studied temperature range. The network was then trained in a similar manner described above, with the normalized coordinates. The quality of output neurons to indicate the C and G states for the testing data is equally good as in the previous case, shown in Fig. 3(c).

While these numerical experiments were performed by using the configurational data generated from GSM, we placed the network into the ultimate test in the third numerical experiment. This time, the NN parameters determined in the second experiment were retained and we asked the network to recognize the configurations generated from the FENE model, in which completely different potential functions were used than in the square-well GSM. Furthermore, instead of producing the configurations from the canonical ensemble where k_BT/ϵ is used as the system parameter, we generated configurations from the Wang-Landau (WL) algorithm for microcanonical ensembles, in which the total (reduced) energy per particle $e = E/N\epsilon$ is directly used as the system parameter. At each e value illustrated in Fig. 3(d), 1×10^3 independent FENE configurations, normalized by their corresponding S,

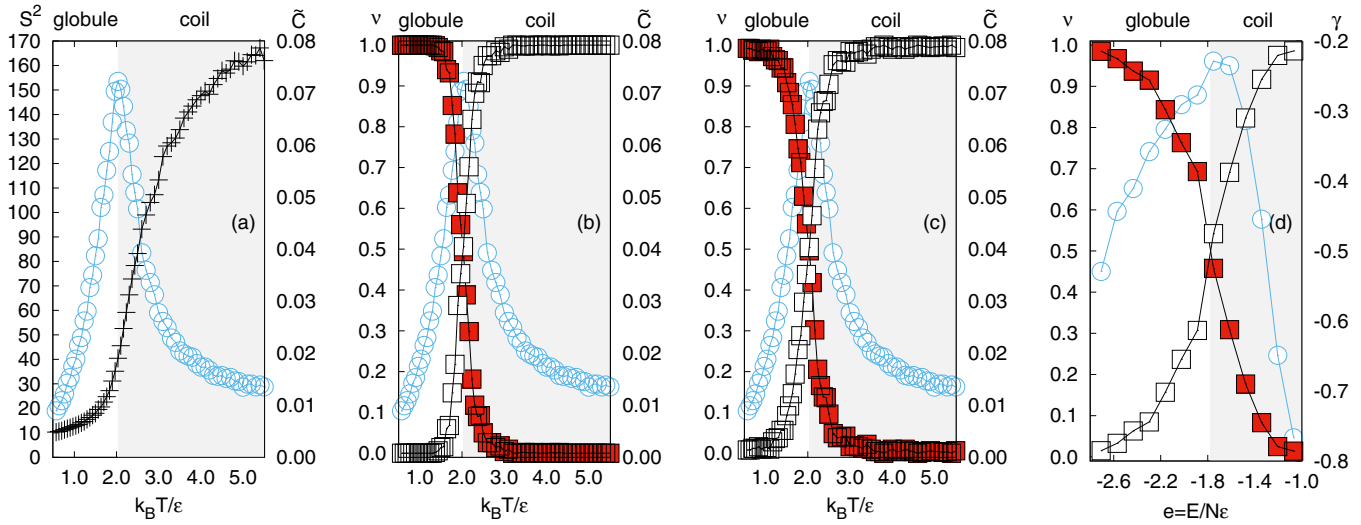

FIG. 3. (a) Reduced mean-square radius of gyration (plus symbols in the left scale) and specific heat (circles in the right scale) as functions of the reduced temperature k_BT/ϵ, determined from the MC simulations of GSM for $N = 102$; (b) the NN outputs of test-recognizing independent GSM configurations; (c) the NN outputs of test-recognizing independent, *normalized* GSM configurations; and (d) the NN outputs of test-recognizing normalized FENE configurations. The filled and open squares represent the mean ν values output from the globular and coil neurons, respectively. The circles in plot (d) represent the specific-heat-like γ (right scale) determined for the FENE model from a WL MC simulation. Error bars associated with all squares are smaller than the symbol size.

were sent to the GSM-trained network for testing. The two recognition curves produced from the output neurons, shown in the figure as the underlying curves behind filled and open squares, predicate a C-to-G transition at $e = -1.8$. This prediction can be independently verified by examining the $\gamma(e)$ curve, which is a specific-heat-type measurement in reduced units defined in the microcanonical ensemble [32]. The data points represented by circles in Fig. 3(d) were calculated based on the density of states determined from the WL algorithm; it shows a peak at the same location as the one successfully predicted by GSM-trained NN.

C. Low-energy polymer states

It is well known that the FENE model exhibits three different states in the low-energy region, globule, anti-Mackey (aM), and Mackey (M), reported by careful MC studies utilizing the WL algorithm which uses e as the system parameter [31,32]. One could convert all languages to a low-temperature description but here we keep the low-energy description. These three structures have subtle structural differences which cannot be distinguished by direct visualization in Figs. 2(b), 2(c), and 2(d). In particular, the crystalline aM and M differ only slightly from each other by the way that monomers are stacked [Figs. 2(f) and 2(g)].

As a final numerical experiment, we challenge the NN to recognize these three states, using three neuron nodes in the output layer, each assigned to recognize G, aM, and M separately. The network was trained with FENE configurations in the energy range $e = [-4.3, -4.16]$, $[-4.7, -4.5]$, and $[-4.9, -4.8]$, where the G, aM, and M structures, respectively, can be clearly defined. The training data contained 3×10^3 configurations at every energy bin.

After the network was adequately trained, it was tasked to recognize an independent set of test data covering the entire energy range in Fig. 4. Over 10^3 configuration samples were used at every energy bin for this purpose. The mean ν values of the test output, from the G, aM, and M nodes, are represented in Fig. 4 by filled squares, open diamonds, and filled diamonds. The intersections of the interpolated curves predict that G-to-aM and aM-to-M transitions take place at $e = -4.40$ and $e = -4.74$, respectively. These NN predicted transition points can be confirmed by an examination of γ, independently calculated from the WL MC simulations. From the γ peaks in the plot, we determine that the G-to-aM and aM-to-M transition points

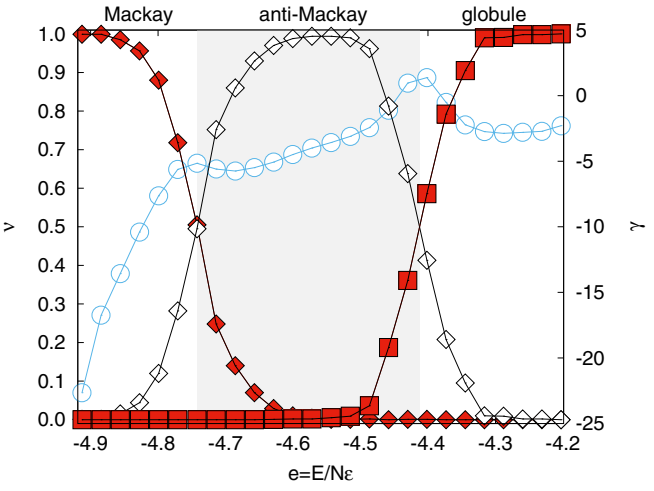

FIG. 4. Mean NN outputs ν (square for globule, open diamonds for anti-Mackay, and filled diamonds for Mackay) from the test samples, after the network is trained to recognizing these states in regimes where they are stable. In the background, the reduced specific heatlike γ (circles in the right scale) was independently produced from the MC simulations. Error bars are smaller than the symbol size.

are at $e = -4.40 \pm 0.03$ and $e = -4.74 \pm 0.03$, respectively, which agree with those from the NN predictions.

D. In search of a phase transition

The above numerical experiments demonstrated the NN's versatility in recognizing polymer configurations and the usefulness of NN in determination of the transition points. There are two essential questions we have not adequately addressed. How does the NN predicted location of the transition depend on the range of used training data? Would the network mistakenly identify a phase transition for a system where a certain physical property smoothly crosses over from relatively large to small values without going through a real phase transition?

To answer the first question we return to the GSM MC data over a wide range of the reduced temperature, [0.5,3.5]. We conducted a series of 18 independent training sessions, each assigned a different training temperature range, away from the C-to-G transition point. The first session treated MC configurations produced from the system at two temperature values, one on the extreme left and the other extreme right, where the C-to-G transition sits in the middle. In the subsequent sessions the training temperature ranges expanded from the extreme left to the center and extreme right to the center, incrementally. These trained networks were then put into test by inputting independent configurations over the extended [0.5,5.5] range. The mean output ν values from the C and G nodes are illustrated in Fig. 5(a) for a few selected sessions. As the training range expands the NN-predicted transition temperature converges to a fixed point; the final converging temperature agrees with the one determined by the MC $\tilde{C}$ peak, 2.03. This study suggests that the approximate location of the transition temperature can be already estimated by using early training ranges far away from the transition point and that the more precise determination can be achieved by progressively adding configurations closer to the transition point. The procedure actually reveals a mechanism of finding a phase transition point, without *a priori* knowledge of its existence, by taking two small training ranges as the starting point and proposing a phase transition point somewhere in between.

To answer the second question, we conduct a numerical experiment on a NN with GSM MC data in the reduced temperature range [2.5,5.5]. Within this coil region, both S^2 and average energy (not shown) have significant variations. We enforcedly train the NN so the two output neuron nodes mistakenly regard configurations in the range [2.5,3.5] as in phase-1 and [4.5,5.5] as in phase-2. We then test the network with independent configurations over the entire [2.5,5.5] range. The results of the output nodes are plotted in Fig. 5(b). Each node vaguely recognizes the configurations as "phase-1" or "phase-2," with a mean ν value hovering around 0.5 in high uncertainties. No clear signals, such as those determined above for true phase transitions, exist.

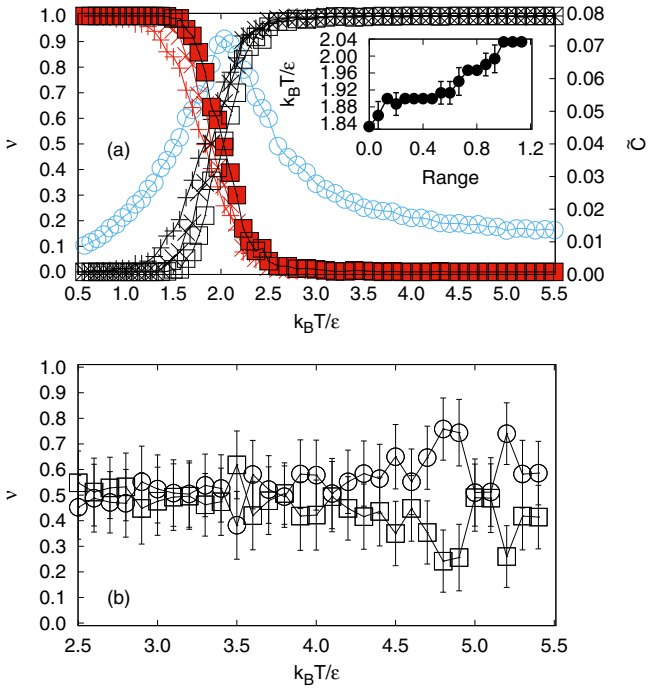

FIG. 5. (a) Average NN output ν from the globule and coil neurons on the testing configurations for NN models trained in various temperature ranges and (b) average NN output ν from "phase-1" (squares) and "phase-2" nodes (circles) on the test configurations in the temperature range [2.5,5.5] for NN models trained by using low- and high-temperature samples. The inset in plot (a) is the NN-predicted transition temperature as a function of the training temperature range used. Error bars in (a) are smaller than the symbol size. The plus, cross, and square symbols represent the mean ν from the two output neurons on test samples, of NN models initially trained in the temperature range, Range = 0.00,0.53, and 1.20, respectively. The circles in (a) represent the same data in Fig. 3(a). The temperature range in (b) contains no actual phase-transition point.

IV. SUMMARY

We described the training of neural networks to recognize diversely and subtly different polymer states produced from Monte Carlo simulations. One advantage of this approach is directly sending the configuration data represented by molecular coordinates to a NN, *without* defining order parameters or calculating the heat capacity; these are conventionally used in computer simulations to rigorously determine a transition point. In particular, in the low-energy regime, a simulated system often encounters potential-energy traps which need to be treated by using a non-Boltzmann weight. Taking the Wang-Landau algorithm as an example, the calculation of the heat capacity in the extremely low-energy regime requires high numerical precision of the computed density of states, which is achievable but requires extensive computations. The NN process describe here, on the other hand, does not require such precision, as long as independent configurations used for supervised training are produced by a numerical simulation.

The example used here is a classical molecular system displaying gas-, liquid-, and crystal-like structures at various energies. We demonstrated that NN can classify both first- (globule to anti-Mackay) and second-order (coil-to-globule and anti-MacKay-to-Mackay) transitions. The direct use of molecular coordinates as input into the NN underlies the robustness and simplicity of our approach and suggests that other simulation tools, such as molecular dynamics, could

be used as well. The outcome of this work provides a compelling reason to incorporate machine-learning techniques into molecular simulations more generally, as a powerful hybridized computational tool for the future study of polymeric systems.

ACKNOWLEDGMENTS

Financial support from the Natural Sciences and Engineering Council of Canada is gratefully acknowledged. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network and Compute Canada. R.G.M. acknowledges support from the Canada Research Chair program and the Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported through Industry Canada and by the Province of Ontario through the Ministry of Research & Innovation.

APPENDIX A: GAUSSIAN-CHAIN MODEL WITH A SQUARE-WELL POTENTIAL

In the paper, the first model we used is a polymer made of monomers that are connected by spring potentials. The reduced bonded Hamiltonian of the polymer is

$$\beta E_{\text{bond}} = \frac{3}{2a^2}\sum_{i=1}^{N-1}(\mathbf{r}_i - \mathbf{r}_{i+1})^2, \tag{A1}$$

where $\beta = 1/k_B T$ is the Boltzmann factor. The system interaction potential energy is

$$E_{\text{int}} = \frac{1}{2}\sum_{ij} U(|\mathbf{r}_{ij}|), \tag{A2}$$

where $\mathbf{r}_{ij}$ is the distance vector between monomer i and monomer j. The pairwise interaction potential has the form

$$U(r) = \begin{cases} \infty & \text{if } 0 \leqslant r^2 < (0.9a)^2, \\ -\epsilon & \text{if } (0.9a)^2 \leqslant r^2 \leqslant 2a^2, \\ 0 & \text{otherwise.} \end{cases} \tag{A3}$$

We refer to this model as GSM in the text.

APPENDIX B: FENE MODEL WITH A LJ INTERACTION

In the text, we used the second polymer model which has a system energy E as the sum of the bonded and interaction energies. The former is described by

$$E_{\text{bond}} = \sum_{i=1}^{N-1} E_{\text{FENE}}(|\mathbf{r}_i - \mathbf{r}_{i+1}|), \tag{B1}$$

where E_{FENE} is a particular realization of the FENE model,

$$E_{\text{FENE}}(r) = -20\epsilon R^2 \ln[1 - (r - r_0)/R)^2], \tag{B2}$$

in which r is the distance between two connected monomers, $R = 0.3b$ controls the bond-length variations, and $r_0 = 0.7b$. The interaction potential energy formally follows (A2) but the two-body interaction is replaced by a truncated Lennard-Jones potential,

$$U(r) = \begin{cases} U_{\text{LJ}}(r) - U_{\text{LJ}}(r_c) & \text{if } 0 \leqslant r < r_c \\ 0 & \text{otherwise,} \end{cases} \tag{B3}$$

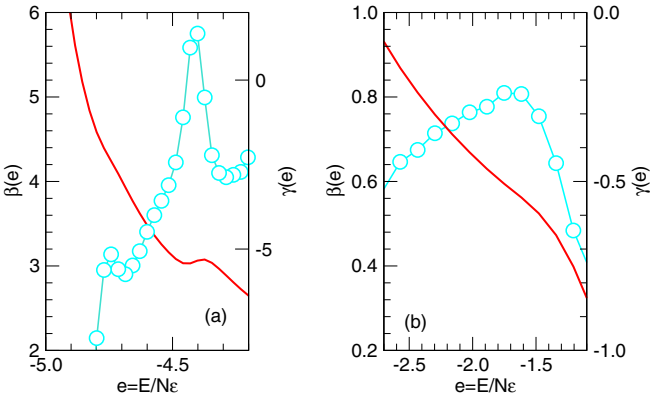

FIG. 6. Inverse temperature $\beta(e)$ [red (dark gray) in the left scale] and its derivative $\gamma(e)$ [light blue, (light gray) in the right scale] as functions of reduced energy per monomer $e = E/N\epsilon$, in (a) low- and (b) midlow-energy regimes. The peaks in the heat-capacity-like $\gamma(e)$ separate different polymer phases: coil, globule, anti-Mackay, and Mackey (from high- to low-e regimes). The FENE model was used in Wang-Landau Monteo Carlo simulations to produce this figure.

where $r_c = 2.5\sigma$. The Lennard-Jones potential has the standard form,

$$U(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{B4}$$

where $\sigma = 2^{-1/6} r_0$ and ϵ measures the potential-well depth (which is adjusted by the value at the truncation point).

The inverse temperature $\beta(e)$ and its derivative $\gamma(e)$ defined in a microcanonical ensemble are shown in Fig. 6, produced from Monte Carlo simulations following the Wang-Landau algorithm [34,35]. The left panel is similar to Fig. 1 in Ref. [32], but shifts in both e and vertical directions are noticeable. These shifts do not affect the physics we discuss in this paper. The current parametrization exactly follows the description in Ref. [32], except for the length scale b included here for accounting purposes; it is unknown where these shifts come from.

APPENDIX C: MC SIMULATION WITH THE BOLTZMANN WEIGHT

Configurations and measurements used in Figs. 3(a), 3(b), 3(c), and 5 in the text were produced from MC runs in which the Boltzmann weight $W = \exp(-\beta E_{\text{bond}} - \beta E_{\text{int}})$, where the two potential energies are expressed in Eqs. (A1) and (A2), were used. Every MC step (MCS) contains N repeated MC trial moves of locally displaced monomers. For every specified $k_B T/\epsilon$, 10^7 MCS was used in the initial equilibration and 5×10^3 configurations were recorded in a production run comprised of 2×10^8 MCS, taken with a lapse of 4×10^4 MCS. The mean-square radius of gyration is defined by:

$$\langle R_g^2 \rangle = \left\langle \frac{1}{N}\sum_{i=1}^{N}(\mathbf{r}_i - \mathbf{r}_{\text{c.m.}})^2 \right\rangle, \tag{C1}$$

where $\mathbf{r}_{\text{c.m.}}$ is the center of mass of a polymer. The reduced specific heat is defined by:

$$\tilde{C} = \left(\left\langle E_{\text{int}}^2 \right\rangle - \langle E_{\text{int}} \rangle^2\right)/\epsilon^2. \quad \text{(C2)}$$

In both cases, the MC average $\langle \ldots \rangle$ was performed in the production run.

APPENDIX D: MC SIMULATION WITH THE WANG-LANDAU ALGORITHM

Using the Wang-Landau algorithm, we determined the density of states of the FENE model, which was used for the calculation of $\beta(e)$ and $\gamma(e)$ (both defined and discussed in Ref. [32]) in Fig. 3(d), Fig. 4 in the text, and Fig. 1 in Appendix B, over a wide range of energy space by conducting two series of simulations, one covering $e = [-5, -4]$ and the other $[-3, 2]$. In total, 30 energy bins were used in each simulation. We used the procedure described in Refs. [34,35], with a final modification-factor $f_{\text{final}} = \exp(2^{-29})$ to produce high-precision data. One small revision is made to the original procedure; when the inverse density of states is used as the MC transition weight, linear interpolations are introduced to connect the logarithmic values of the density of states at the centers of adjacent energy bins; this is in contrast to the original histogram-type weight scheme.

Configurations used in these figures were produced from a production run consisting of 10^9 MCS, in which the inverse density of states was used as the MC transition weight. Approximately 5×10^3 configurations were recorded at every energy bin.

[1] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, AI Mag. **13**, 57 (1992).
[2] T. L. H. Watkin, A. Rau, and M. Biehl, Rev. Mod. Phys. **65**, 499 (1993).
[3] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques* (Morgan Kaufmann, San Francisco, CA, 2005).
[4] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach* (Springer Science & Business Media, Berlin, 2013).
[5] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, Cambridge, MA, 2012).
[6] M. Jordan and T. Mitchell, Science **349**, 255 (2015).
[7] K. Fukushima, Neural Netw. **1**, 119 (1988).
[8] H. S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992).
[9] R. Parekh, J. Yang, and V. Honavar, IEEE Trans. Neural Netw. **11**, 436 (2000).
[10] A. K. Jain, R. P. W. Duin, and J. Mao, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 4 (2000).
[11] C. Davatzikos, K. Ruparel, Y. Fan, D. Shen, M. Acharyya, J. Loughead, R. Gur, and D. D. Langleben, Neuroimage **28**, 663 (2005).
[12] C. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2007).
[13] J. Schmidhuber, Neural Netw. **61**, 85 (2015).
[14] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard *et al.*, Neural Netw. **261**, 276 (1995).
[15] M. A. Nielsen, http://neuralnetworksanddeeplearning.com/.
[16] R. P. Lippmann, Neural Comput. **1**, 1 (1989).
[17] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, IEEE Signal Process. Mag. **29**, 82 (2012).
[18] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, Phys. Rev. Lett. **108**, 253002 (2012).
[19] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[20] P. Geiger and C. Dellago, J. Chem. Phys. **139**, 164105 (2013).
[21] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Phys. Rev. B **90**, 155136 (2014).
[22] J. Carrasquilla and R. G. Melko, Nat. Phys., doi:10.1038/nphys4035.
[23] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).
[24] L. Huang and L. Wang, Phys. Rev. B **95**, 035105 (2017).
[25] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 041101 (2017).
[26] N. Portman and I. Tamblyn, arXiv:1611.05891 (2016).
[27] P. Mehta and D. J. Schwab, arXiv:1410.3831 (2014).
[28] H. W. Lin and M. Tegmark, arXiv:1608.08225 (2016).
[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, J. Mach. Learn. Res. **15**, 1929 (2014).
[30] S. Schnabel, T. Vogel, M. Bachmann, and W. Janke, Chem. Phys. Lett. **476**, 201 (2009).
[31] D. T. Seaton, T. Wüst, and D. P. Landau, Phys. Rev. E **81**, 011802 (2010).
[32] S. Schnabel, D. T. Seaton, D. P. Landau, and M. Bachmann, Phys. Rev. E **84**, 011127 (2011).
[33] M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics* (Oxford University Press, New York, 1986).
[34] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).
[35] F. Wang and D. P. Landau, Phys. Rev. E **64**, 056101 (2001).