Data-Driven Models for Predicting Intrinsically Disordered Protein Polymer Physics Directly from Composition or Sequence

Tzu-Hsuan Chao,[1] Shiv Rekhi,[2] Jeetain Mittal,[2] and Daniel P. Tabor[1]

[1] *Department of Chemistry, Texas A&M University, College Station, TX 77843 USA*
[2] *Department of Chemical Engineering, Texas A&M University, College Station, TX 77843 USA*

(Dated: March 23, 2023)

The molecular-level understanding of intrinsically disordered proteins is challenging due to experimental characterization difficulties. Computational understanding of IDPs also requires fundamental advances, as the leading tools for predicting protein folding (e.g., Alphafold), typically fail to describe the structural ensembles of IDPs. The focus of this paper is to 1) develop new representations for intrinsically disordered proteins and 2) pair these representations with classical machine learning and deep learning models to predict the radius of gyration and scaling exponent of IDPs. Here, we build a new physically-motivated feature called the bag-of-amino-acid-interactions, which encodes pairwise interactions explicitly into the representation. This feature essentially counts and weights all possible non-bonded interactions in a sequence and thus is, in principle, compatible with arbitrary sequence lengths. To see how well this new feature performs, both categorical and physically-motivated featurization techniques are tested on a computational dataset containing 10,000 sequences simulated at the coarse-grained level. The results indicate that this new feature outperforms the others and possesses solid extrapolation capabilities. For future use, this feature can potentially provide physical insights into amino acid interactions including their temperature-dependence, and be applied to other protein spaces.

I. INTRODUCTION

Many intrinsically disordered proteins (IDPs) are biologically active even though they do not possess well-defined folded structures.[1,2] IDPs are characterized by a free energy landscape with multiple local minima, leading to an ensemble of configurations as opposed to a global minimum seen in the case of proteins with stable folded structures.[3,4] Characterizing the structure of these highly dynamic and flexible proteins is challenging, both experimentally and through simulations, due to the low free energy barriers[5,6] and numerous relevant conformations.[3] These challenges have led to an approach of connecting measurable ensemble average properties, such as radius of gyration (R_g), to ensembles through polymer physics theory.[7–13] However, the prediction of such conformational properties directly from primary sequences for such disordered proteins remains highly challenging, relying on combinations of experimental, theoretical, and simulation methods.[14–17]

Machine learning methods have emerged as a valuable tool in uncovering the relationship between primary sequences of proteins, as well as nucleic acids, and their properties.[18–22] The goal of these models is to encode the sequence (using categorical features) and correlate these features with the relevant physical observables. With these low-cost models, the goal is to design new sequences or materials with a desired output at a much lower cost than experimental or physics-based modeling approaches (Fig. 1a). In the context of proteins, machine learning has been applied extensively to the prediction of the higher-order structure based on only the primary structure of the protein.[23–27] Such approaches are able to predict regions of proteins that would remain disordered. Additional machine learning efforts have focused on generalized coarse-grained polymer models. Machine learning models which take in physically motivated features that are analogous to color mapping have been built based on parameters from coarse-grained models and have been demonstrated to effectively predict the average properties of these polymers.[28] A limitation of these methods is that the dimensions of input features scales with the protein length, leading to potential difficulties in training, and requiring foreknowledge of the maximum length sequence that a model will need to consider.

In this work, we introduce a new feature named the "bag of amino acid interactions" representation (BAA), which accounts for the pairwise non-bonded interactions of amino acids present in the sequence. The dimensions of this feature remain constant at 400 (20×20 amino acids), irrespective of the length of the protein being investigated. Additionally, by considering the importance of order in a sequence, the bag of amino acid interactions feature can differentiate between sequences of the same amino acid composition, but with different sequencing of residues.

To compare the performance of this representation in relation to others, we considered a data set of coarse-grained simulations of randomly generated sequences (referred to as the IDP-10260 dataset in this work)[29] using an implicit solvent, one-bead-per-residue model that has been previously shown to accurately predict the average radius of gyration (R_g) of IDPs.[30] We use these simulation results as the prediction targets to train models with categorical[22,31] and physically-motivated features. We test the performance of these features for both classical machine learning models and artificial neural networks, where appropriate, for each feature. For the dataset considered in this work, we find that the bag of amino

acid features marginally outperforms other features while maintaining computational efficiency and introducing desirable characteristics of future use cases. In addition, we present a set of extrapolation tests that provide some insights into the degree that these model-feature combinations can be tested outside of their training domains, which we consider to be an important test due to the potential use of machine learning models in design tasks.

This paper is organized as follows. First, we describe the computational methods to used to generate the training and test data. We then discuss the featurization strategies considered in this work, including categorical features and physically-motivated features, and the machine learning models these features are combined with. Then, we present our results and discussion, including a test of the extrapolation ability of BAA representation and the integration of the feature with various temperature data.

II. THEORETICAL AND COMPUTATIONAL METHODS

A. Simulation details for polypeptide training and testing data

All of the simulations on the IDP-10260 dataset were conducted using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS).[32] The initial configuration of the polypeptides is a linear chain positioned in the middle of a cubic box which has a side length of 1500 angstroms. For each sequence, seven simulations are performed under different temperatures in an NVT ensemble for 500 nanoseconds with a time step of 10 femtoseconds. The radius of gyration is calculated from snapshots between 100 to 500 ns. More simulation details and example simulation snapshots are provided in the supporting information (Section S6).

B. Featurization

1. Categorical Features

Categorical features are the dominant features[19,33–35] used in data-driven protein models. This featurization strategy is employed in both supervised and unsupervised learning. Prior work has shown that categorical features can be used for accurate property or classification models when enough data (7000 points) is fed into the model.[36] For proteins, these sequences can be expressed using letters, which are then transformed into a vector. With categorical features, we focus on two traits: 1) if the features are size-explicit or size-implicit and 2) if the features are order-dependent or order-independent. Recent work has shown that size-implicit features perform similarly to size-explicit features.[31] Considering important differences that may arise due to the size and order information explicit or implicit in an encoding, we used three categorical features in this work (key traits, summarized in Fig. 1b): 1) Count encoding (size-implicit and order-independent), 2) Ordinal encoding (explicit length information and order-dependent), 3) One-hot encoding (size-implicit and order-dependent).

Count Encoding (CE) The count encoding (CE) feature encodes the number of occurrences of a given amino acid in a sequence, similar to a character count in a word. In this work, the count encoding is a 20-element array, where the value of each index indicates the number of a given amino acid in the sequence. We note that this feature is size-implicit since the sequence size can be calculated by the sum of the array.

Ordinal Encoding (OE) The ordinal encoding (OE) feature encodes the sequence into a finite length array (in this case the length is 200, the largest sequence considered in the training and testing set), where each index is assigned an integer that corresponds to one of the two amino acids. To ensure the dimensionality of the feature is the same for the whole dataset (necessary for compatibility with many ML models), those sequences shorter than the longest one in the data will pad zero at their ends. For instance, the ordinal encoding inputs used in this paper have 200 features, which is the largest sequence in this data set.

One-Hot Encoding (OHE) The one-hot encoding (OHE) feature encodes the sequence in an array consisting of only 0 or 1 to represent the absence or presence of the amino acid at the position. For a sequence with length 200, there will be 200×21 elements in the one-hot encoding feature. In each position, there are 21 elements. The i^{th} element is one if the i indexed amino acid is present. The 21^{st} element is one if no amino acid at that position. One-hot encoding is size-implicit and ordered. The sum of the array from 1^{st} to 20^{th} element at each position corresponds to the sequence size. Although this feature keeps the size and order information, it has many more dimensions than the other two categorical features. Higher dimensions could not only cause overfitting but also require longer training times and larger sets to train. The inclusion of this feature was motivated by the results of Ref. 22, where the combination of one-hot encoding and a linear-regularized regression model was shown to perform well for evolutionary sequence data prediction tasks.

2. Physically-motivated Features

Color Mapping The color mapping feature representation is derived based on its original description in Ref. 28, where it was applied to predicting the radius of gyration calculated from coarse-grained polymer simulations. Rather than using the index number of an amino acid as in the ordinal encoding, the index number is replaced by the coarse-grained simulation parameters such as hydropathy, charge, and the particle's van der Waals radius.

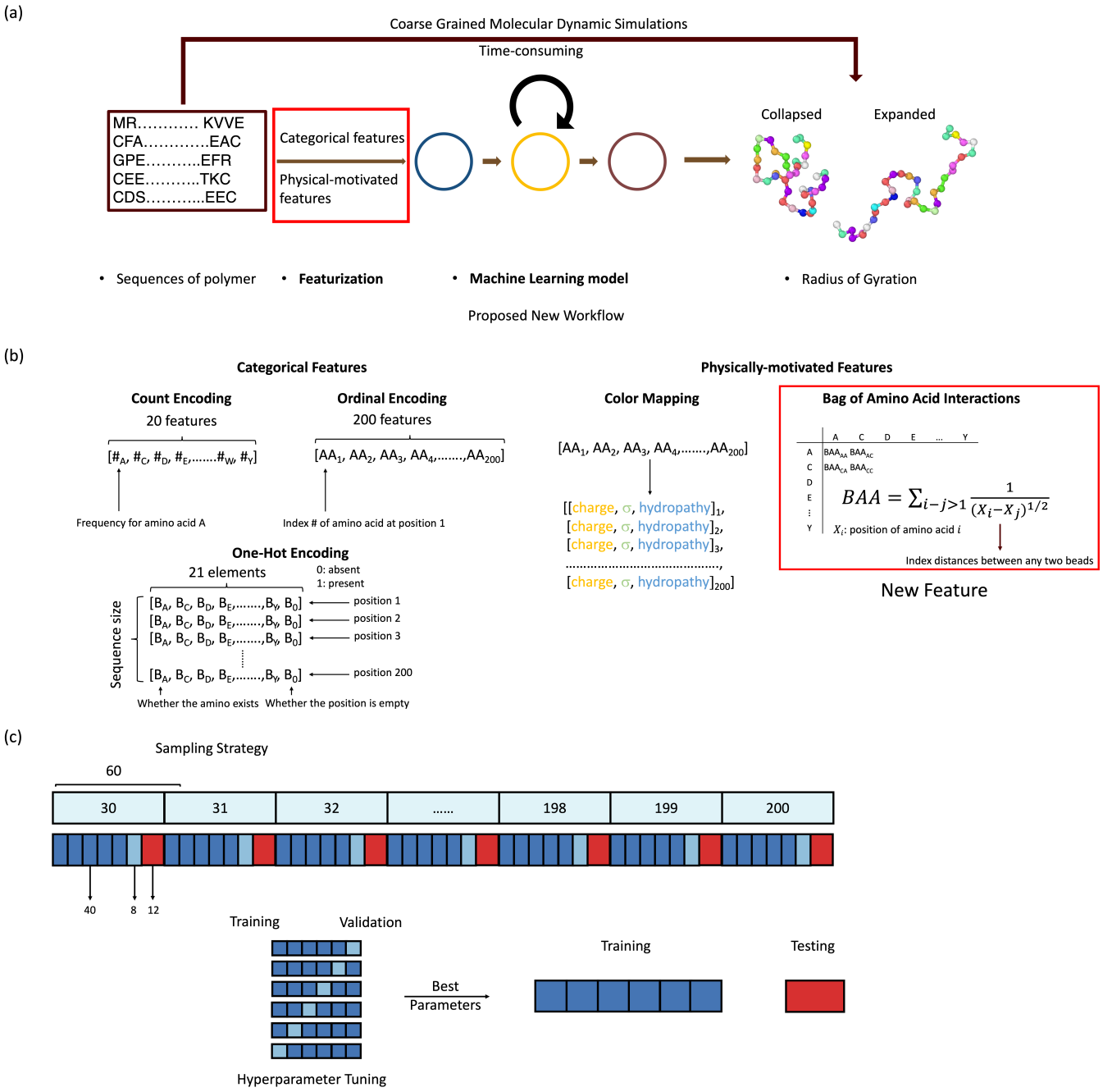

FIG. 1. Featurization techniques overview. (a) Overall workflow for predicting IDP properties directly from primary sequences. In this work, we are focusing on featurization techniques. (b) Graphical illustration of the categorical and physically-motivated features considered in this work (c) The sampling strategy employed to train the machine learning models, based on the composition of the benchmarking set, has equally distributed chain lengths. We sample the data equally and perform cross-validation.

Thus, for the sequences in this study, the total number of dimensions for the input is 600. The feature can be either remain as a 2D array or be flattened into a 1D array and fed to Convolutional 2D or 1D neural networks, as discussed further below.

Bag of Amino Acid Interactions (BAA) The BAA feature developed for this work was motivated by the importance of pairwise interactions[37,38] in determining the properties of disordered proteins.[39] Some efforts have been made on predicting the contact density and potential in a sequence[40]. However, those methods still encounter the scaling problem when the sequence size gets larger and use an arbitrary cutoff for feature definitions. Herein, inspired by the sequence charge decoration[41] and

sequence hydropathy decoration parameters,[29] we developed a new feature called bag of amino acid interactions. In this feature, the pairwise interaction counts are calculated by how many specific pairs of non-adjacent amino acids are in the sequence. Each count is divided by the square root of the number of bonds between each amino acid in that pair. The feature will be a 20×20 array, totaling 400 features, and will not increase for longer sequences. A demonstration of converting an example sequence into each of the physically-motivated features is shown in the supporting information (Fig. S4).

C. Machine Learning Models

The machine learning models used in this paper can be classified into classical models and artificial neural network models. In general, classical models are used when limited training data is available and artificial neural networks are used when a larger dataset is available. However, each model may find use outside of its normal context, and the performance of each model for a particular feature can aid in the interpretability of the models and give insights into further feature development. Here, we provide a brief description of the machine learning models used in this work and highlight prior applications to protein property prediction models, where appropriate.

L2-Linear Regularized Regression (LRR) A linear regression model is the most straightforward model, which employs a linear model on the features. To train the model, the loss is minimized between the predicted values and target values (in the case of an L2 model, the sum of the square of the losses). To avoid overfitting, a regularization term is added to the loss function, which is the sum of the coefficient in the functions. Larger regularization terms indicate more regularized models. In Ref. 22, this technique was used in combination with one-hot encoding.

Kernel Ridge Regression (KRR) Kernel ridge regression is a regularized model using a kernel trick to transform the target function into desired forms, which is not restricted to a linear model. The kernel applied in this paper is the radial basis function.

$$g(\theta) = \sum_{n=1}^{N} (y^{(n)} - f(s^{(n)};\theta))^2 + \lambda \sum_{i=1}^{L} \sum_{a \in A} \theta_i(a)^2 \quad (1)$$

Support Vector Regression (SVR) Support vector regression uses a different strategy than LRR to calculate the loss function. Rather than calculate the loss function directly, a hyperplane is added and any errors for data points within the range will be considered as 0. For data points outside the hyperplane, the loss function is the error - hyperplane width ϵ. The kernel is the radial basis function.

Gaussian Process Regression (GPR) Gaussian process regression starts with a prior probability distribution. It uses a covariance kernel function and Bayes' rule to acquire a posterior probability function.

Feed-forward Neural Network (FNN) The first type of neural network in this work is a simple uni-directional neural network, where each layer is fully connected to the next layer.

Convolutional Neural Network (CNN) The second type of neural network in this work is chosen due to its potential compatibility with the "color mapping" feature representation. In this case, the "RGB" values for the convolutional neural network are the coarse-grained parameters. Given the structure of the data (sequences), a one-dimensional convolutional model is applied here. The output from the last convolutional layer is flattened and fed into fully connected hidden layers and the final outputs are scaling exponent and radius of gyration.

D. Training Process

For the training and testing process, we the IDP-10260 dataset, which was randomly generated from a distribution using all of the amino acids.[29] It consists of 171 chain lengths and 60 data points for each chain length (total 10260). To ensure the training quality, the training set is sampled to have an equal number of data points from each chain length. For the classical model hyperparameter tuning, 80% is used to undergo a six-fold cross-validation. The remaining 20% is used as the testing set. For generating the learning curves, eight splits are applied in each fold. For the neural network models, 60% is used as the training set, 20% is used as the validation set and the model is trained based on the validation loss.

III. RESULTS AND DISCUSSION

A. Classical models and categorical features

We consider the performance of the models on predicting both R_g and the scaling exponent. The initial test includes three features (CE, OE and OHE) using four classical machine learning models and the 300 K data to see the correlation between these features and the scaling exponent without the effect of temperature. The testing results are shown in Fig. 2. The R^2, MSE, and RMSE are listed in Table 1. The order of performance is CE > OHE > OE. The combination of CE and SVR gives the best performance. This result indicates that the variation of amino acid compositions can easily capture the structural characteristics of IDPs in the IDP-10260 dataset and that sequence-specific information is not needed for this dataset.

For the OE representation, the model performances for each of the regression methods are similar. The index number used in the feature could potentially interfere

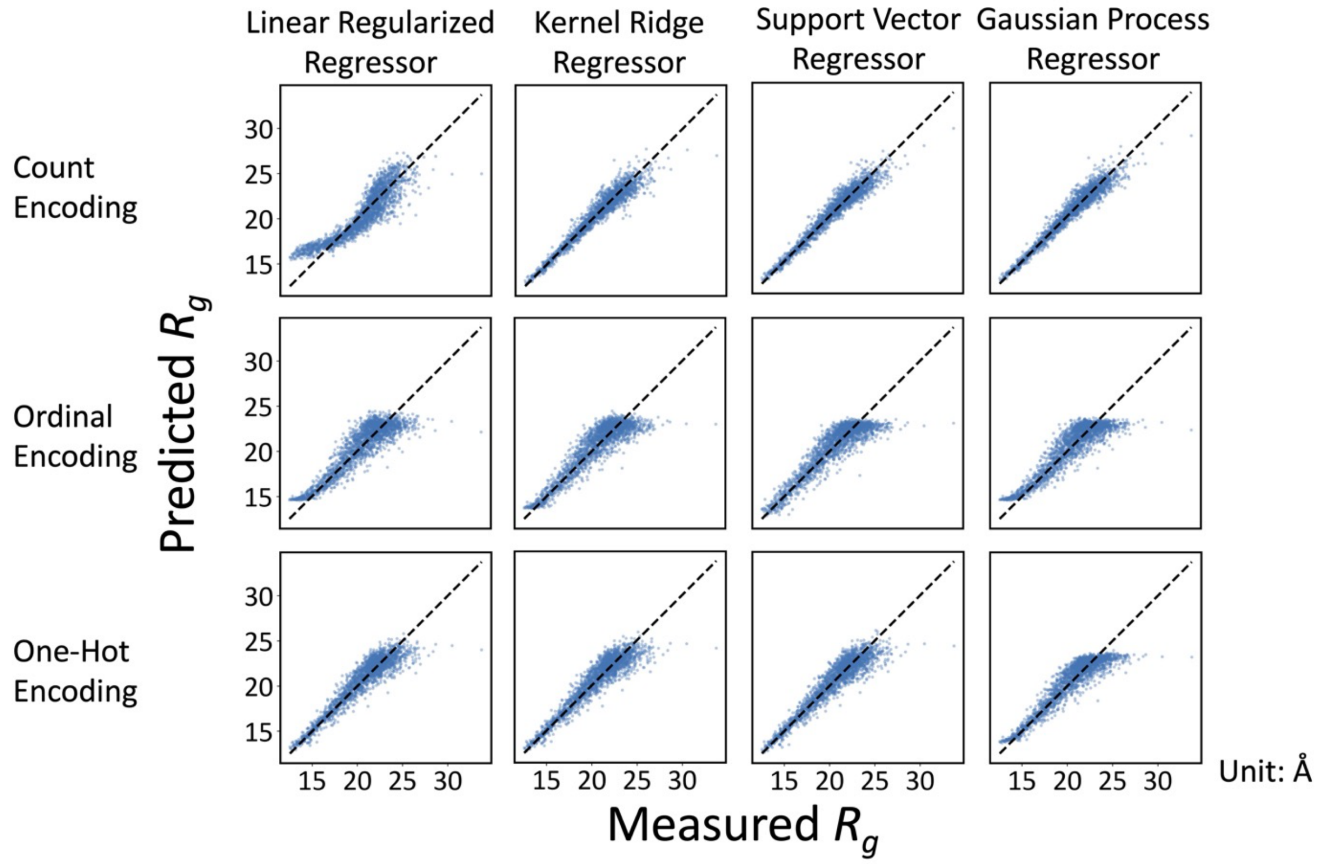

FIG. 2. Categorical feature results. We use four types of classical models to benchmark the categorical features. The best model is shown on the right upper corner, the combination of Support Vector Regressor and Count Encoding.

with the learning process since the index numbers do not correlate with the amino acid properties directly. The performance of one-hot encoding is shown in Fig. 2. The learning curves (Fig. S5-16) indicate that this model is overfitting because of the high dimensionality of this feature. It has 4200 dimensions and many of the elements are zero. This points out the problem that one-hot encoding may suffer from a scaling problem when longer sequences are considered.

To find the most important factor in categorical features, a principal component analysis (PCA) analysis was performed on the count encoding feature. Surprisingly, there is still 60% of the cumulative covariance remaining when one component is left after the dimension reduction. We interpret this result as consistent with recent studies, which have been shown that chain length is the most important information in the CE feature.[28] To test this, we trained a model using only chain length as a feature and compared it with PCA-reduced feature performance. The testing result is shown in Fig. 3a. The performance and data distribution are both very similar. This result suggests that the most important information in the CE feature is the chain length. A that uses chain length as an input feature has an R^2 of 0.77, which can be considered to be a baseline for predicting R_g. While the performance of count encoding is already high, this feature doesn't take sequence order into account and does not further physical insights. In addition, these results may be dataset-dependent, as it is possible to construct sequences with identical compositions but drastically different scaling behavior and values for R_g.[39,42] Thus, we seek to test physically-motivated features that can take orders into account and avoid the scaling problem for future application to longer sequences.

B. Color Mapping

For the color mapping featurization, the testing results (when paired with convolutional neural networks) are shown in Fig. 3b. These results can be directly compared to ordinal encoding, and the improvements in performance from ordinal encoding to color mapping indicate that the inclusion of the coarse-grained parameters may make the feature more physically-informed, compared to having a categorical index number. Moreover,

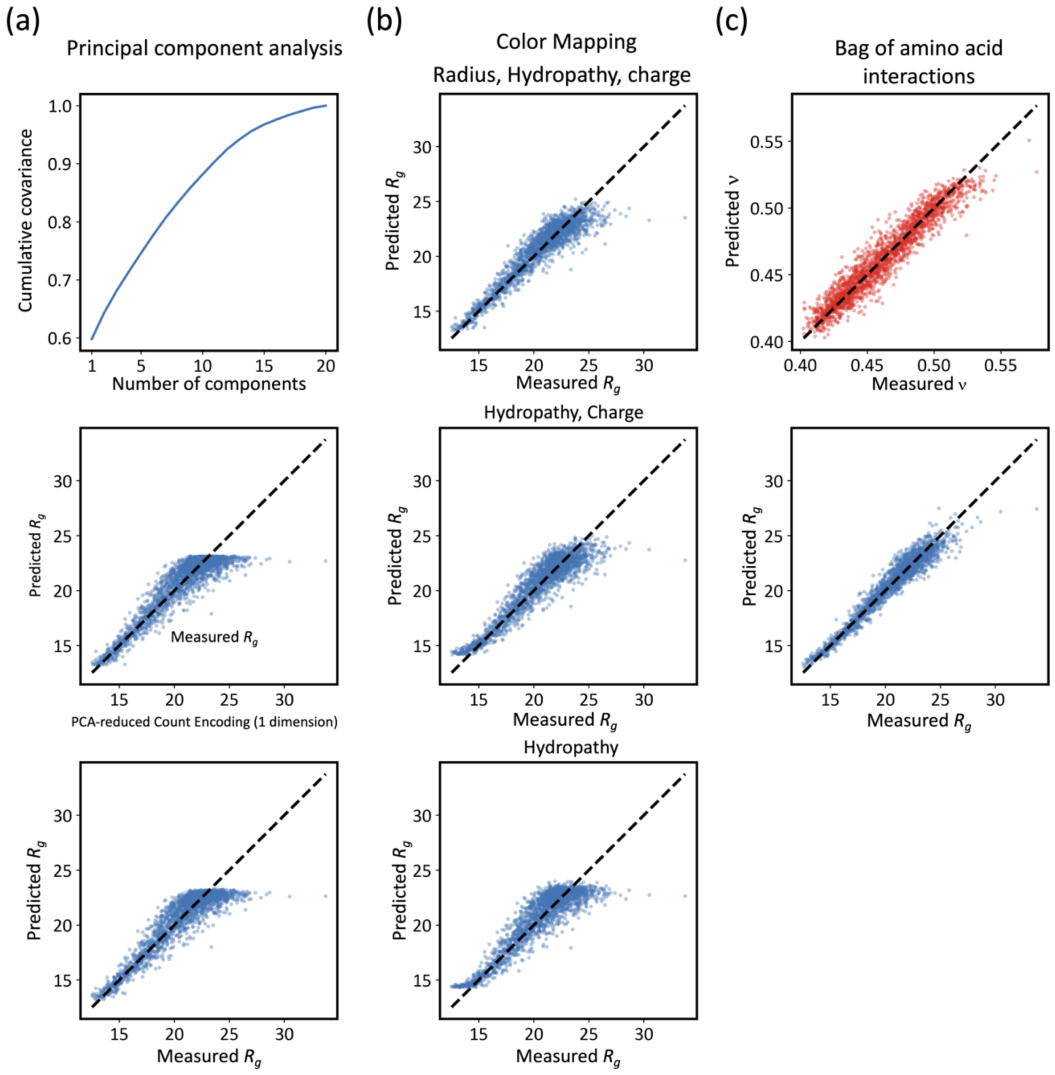

FIG. 3. (a) PCA on the performance of count encoding. The testing results using chain length as the only feature and PCA-reduced Count Encoding give similar performance. (b) Color Mapping testing results. (c) Bag of amino acid interactions testing results, tested on both scaling exponent and the radius of gyration.

we examined the significance of the three coarse-grained parameters on the prediction accuracy. Of the three features, hydropathy turns out to be the most important parameter, with charge as the second most important.

We used both 1D and 2D convolutional layers. Our results show that a 1D convolutional neural network performs better while the original paper used a 2D convolutional neural network (Fig. S17). The benefit of using a 1D convolutional neural network is that the feature extraction process becomes more flexible. Although the overall performance is not better than count encoding, finding the importance of different parameters can be potentially incorporated into a graph neural network that only takes the most significant parameter as the value in each node.

C. Bag of Amino Acid Interactions

The results for the BAA representation are shown in Fig. 3c. We show the results for a model trained using support vector regression (which outperforms other models in categorical features). On this IDP-10260 dataset, the performance is almost the same as count encoding. Though the performance is similar in the aggregate, the models do not have identical results on a sequence-by-sequence basis. Instead, models with similar overall performance arrive at these performance levels through different means. Although the length of the sequence is highly correlated with the R_g (and scaling exponent) in the IDP-10260 dataset, the BAA representation is able to achieve similar performance, despite having no explicit considerations of size in the representation.

TABLE I. Testing R-square, mean squared error(MSE) and root mean squared error(RMSE)

R^2 MSE RMSE	Linear Ridge Regressor	Kernel Ridge Regressor	Support Vector Regressor	Gaussian Process Regressor
Count Encoding	0.80	0.93	0.94	0.94
	1.76	0.61	0.52	0.52
	1.32	0.78	0.72	0.72
Ordinal Encoding	0.80	0.82	0.82	0.81
	1.72	1.61	1.61	1.65
	1.31	1.27	1.27	1.28
One-Hot Encoding	0.88	0.88	0.88	0.86
	1.08	1.07	1.07	1.28
	1.04	1.04	1.03	1.13

There are several advantages of BAA interaction representation. Unlike the other representations considered in this work, the BAA interactions will have the same dimension for longer sequences, and the order of the amino acids is still taken into account. In addition, our formulation of the representation, with an exponent of 1/2 in each of the terms, is one of many possible scaling exponents. We conducted a test of the performance of the model over a range of exponents, and find that many have similar performance (supporting information, Fig. S18 and S19). The flexibility of the model enables each of these interaction exponents to be potentially tuned in future work, when larger (and more challenging) testing sequences are used to train the models. Both the exponents and the weights that are found when the representation is trained with either classical machine learning models or simple neural network models can be connected more deeply to the polymer physics of these sequences and their interactions, particularly when the trained models are applied to simpler model sequences.

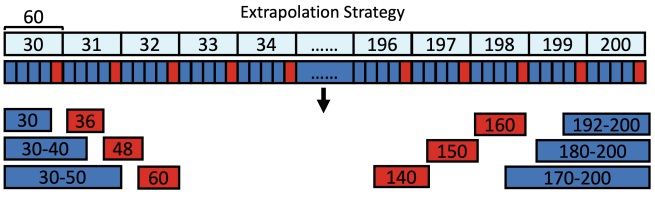

FIG. 4. The extrapolation strategy. The forward process is using 20% larger chain length data for testing while the reverse process is using 20% smaller chain length data as the testing set.

D. Extrapolation Performance

Given the promising initial results of the BAA representation, we developed further tests to examine the ability for the model to perform on out-of-training-sample property prediction tasks. When machine learning models are used in design tasks or in high-throughput virtual screening scenarios, they are often tasked with extrapolating out of their initial training spaces, at least until enough new data has been obtained for retraining. Thus, we conducted two types of tests to determine the extrapolation ability of models that use the new BAA features. Since the most variant factor in the IDP-10260 dataset is the chain length, we do an extrapolation test using either shorter sequences as the training set and longer sequences (20% longer) as the testing set (or *vice versa*). The splitting strategy is represented in Fig. 4. We tested the performance of both support-vector machine and feed-forward neural network models. As seen in Fig. 5a, the support vector regression gives better performance when limited data is applied while the feed-forward neural network requires more data to reduce the test loss. Overall, the performance of the bag of amino acid interaction for longer (or shorter) sequences can still maintain a reasonable test loss.

E. Integrating Temperature into Features

Finally, we tested the compatibility of the BAA model over multiple temperatures. Amino acid pairwise interactions demonstrate temperature-dependent effects that lead to phenomena such as temperature-induced collapse[43] and cold denaturation,[44] to name a few. In order to predict the temperature dependence of IDP size, such temperature-dependent interactions must be encoded within the model. Previous efforts to capture this temperature dependence have relied on empirical temperature-dependent scaling of pairwise interactions[45,46] In this work, we test the compatibility of the BAA model in predicting dimensions of IDPs from sequence over a range of temperatures. Due to the nature of the CG model used, however, we only consider the expansion of the chain with increasing temperature implying that interactions are getting weaker with increased temperature.

Here, we calculated R_g and the scaling exponent at seven different temperatures in the IDP-10260 dataset (270K, 300K, 330K, 360K, 390K, 420K, 450K). By adding one more feature, which is the temperature (in Kelvin), a model was built to predict the scaling exponent at different temperatures. To make the training process easier, we do joint training on both scaling exponent and radius of gyration using a feed-forward neural network. The testing result is shown in Fig. 5b. The R^2 for scaling exponent and radius of gyration are 0.95 and 0.98, respectively. The integration of temperature into the features using the BAA model provides a promising method to further investigate the effect of temperature on the dimensions of IDPs with the potential to further extend the model to account for temperature-induced collapse in the future.

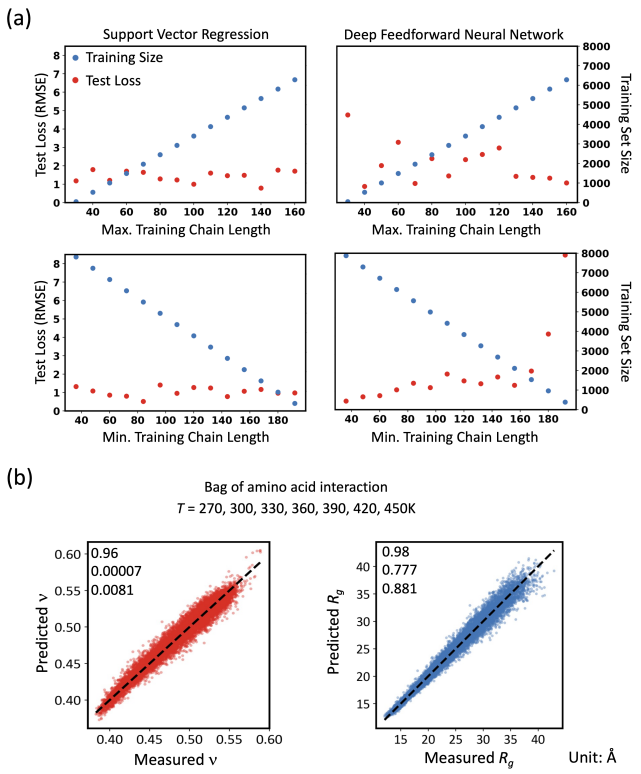

FIG. 5. (a) Extrapolation test results. The extrapolation testing RMSE for both forward extrapolation and reverse extrapolation. The support vector regression gives better results when the training size is small. (b) Testing results including the temperature for ν (left) and R_g (right).

IV. CONCLUSION

In this paper, we built categorical and physically-motivated features and coupled these features with machine-learning models to predict the polymer physics on the IDP-10260 dataset for intrinsically disordered proteins directly from primary sequences. We compared the performance of count encoding, ordinal encoding, and one-hot encoding combined with four classical regression models, and found that count encoding is able to predict both the R_g and scaling exponents with high accuracy. A PCA analysis of count encoding confirmed that chain length is one of the most important factors for predicting scaling exponent within this dataset. The bag of amino acid representation is able to predict both scaling exponent and radius of gyration, is compatible with both classical and artificial neural network models, and does not have an explicit specification of the length in its features. The architecture of the bag of amino acid interactions makes it promising for future applications, as the feature's dimensionality does not increase when longer sequences are introduced. We further tested the extrapolation ability and generalizability for reverse sequences and incorporation with temperature data. In all tests, the representation shows promising accuracy, giving the representation potential for future use as a surrogate model for high-throughput IDP simulation tasks.

CONFLICTS OF INTEREST

There are no conflicts to declare.

ACKNOWLEDGEMENTS

T.-H.C. and D.P.T. acknowledge support from the Robert A. Welch Foundation (Grant No. A-2049-20200401). This research was also partially supported by the Robert A. Welch Foundation (Grant No. A-2113-20220331) and the National Institute of General Medical Science of the National Institutes of Health under the grant R01GM136917.

[1] R. Van Der Lee, M. Buljan, B. Lang, R. J. Weatheritt, G. W. Daughdrill, A. K. Dunker, M. Fuxreiter, J. Gough, J. Gsponer, D. T. Jones *et al.*, *Chem. Rev*, 2014, **114**, 6589–6631.

[2] A. R. Camacho-Zarco, V. Schnapka, S. Guseva, A. Abyzov, W. Adamski, S. Milles, M. R. Jensen, L. Zidek, N. Salvi and M. Blackledge, *Chem. Rev.*, 2022, **122**, 9331–9356.

[3] M. Brucale, B. Schuler and B. Samorì, *Chem. Rev.*, 2014, **114**, 3281–3317.

[4] G. A. Papoian, *Proc. Natl. Acad. Sci.*, 2008, **105**, 14237–14238.

[5] A. B. Oliveira Junior, X. Lin, P. Kulkarni, J. N. Onuchic, S. Roy and V. B. Leite, *J. Chem. Theory Comput.*, 2021, **17**, 3178–3187.

[6] M. R. Jensen, M. Zweckstetter, J.-r. Huang and M. Blackledge, *Chem. Rev.*, 2014, **114**, 6632–6660.

[7] G. L. Dignon, W. Zheng, R. B. Best, Y. C. Kim and J. Mittal, *Proc. Natl. Acad. Sci.*, 2018, **115**, 9929–9934.

[8] H. Hofmann, A. Soranno, A. Borgia, K. Gast, D. Nettels and B. Schuler, *Proc. Natl. Acad. Sci.*, 2012, **109**, 16155–16160.

[9] F. E. Thomasen and K. Lindorff-Larsen, *Biochem. Soc. Trans.*, 2022, **50**, 541–554.

[10] Y. Zhao, R. Cortes-Huerto, K. Kremer and J. F. Rudzinski, *J. Phys. Chem. B*, 2020, **124**, 4097–4113.

[11] G.-N. W. Gomes, M. Krzeminski, A. Namini, E. W. Martin, T. Mittag, T. Head-Gordon, J. D. Forman-Kay and C. C. Gradinaru, *J. Am. Chem. Soc.*, 2020, **142**, 15697–15710.

[12] A. H. Mao, N. Lyle and R. V. Pappu, *Biochem. J.*, 2013, **449**, 307–318.

[13] D. Moses, F. Yu, G. M. Ginell, N. M. Shamoon, P. S. Koenig, A. S. Holehouse and S. Sukenik, *J. Phys. Chem.*

Lett., 2020, **11**, 10131–10136.
[14] J. Huihui and K. Ghosh, *Biophys. J.*, 2021, **120**, 1860–1868.
[15] F. Pesce and K. Lindorff-Larsen, *Biophys. J.*, 2021, **120**, 5124–5135.
[16] L. M. Pietrek, L. S. Stelzl and G. Hummer, *Curr. Opin. Struct. Biol.*, 2023, **78**, 102501.
[17] J. J. Alston, A. Soranno and A. S. Holehouse, *Methods*, 2021, **193**, 116–135.
[18] Y. Yang, M. Zheng and A. Jagota, *Npj Comput. Mater.*, 2019, **5**, 3.
[19] D. M. Varghese, A. Arya and S. Ahmad, *Machine Learning in Bioinformatics of Protein Sequences: Algorithms, Databases and Resources for Modern Protein Bioinformatics*, World Scientific, 2023, pp. 129–151.
[20] W. Khan, F. Duffy, G. Pollastri, D. C. Shields and C. Mooney, *PLoS One*, 2013, **8**, e72838.
[21] X. Liu, *arXiv preprint arXiv:1701.08318*, 2017.
[22] C. Hsu, H. Nisonoff, C. Fannjiang and J. Listgarten, *Nat. Biotechnol.*, 2022, **40**, 1114–1122.
[23] D. E. Kim, D. Chivian and D. Baker, *Nucleic Acids Res.*, 2004, **32**, W526–W531.
[24] L. A. Abriata, G. E. Tamò and M. Dal Peraro, *Proteins: Struct. Funct. Bioinform.*, 2019, **87**, 1100–1112.
[25] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland *et al.*, *Nature*, 2020, **577**, 706–710.
[26] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, *Nature*, 2021, **596**, 583–589.
[27] R. Chowdhury, N. Bouatta, S. Biswas, C. Floristean, A. Kharkar, K. Roy, C. Rochereau, G. Ahdritz, J. Zhang, G. M. Church *et al.*, *Nat. Biotechnol.*, 2022, **40**, 1617–1623.
[28] M. A. Webb, N. E. Jackson, P. S. Gil and J. J. de Pablo, *Sci. Adv.*, 2020, **6**, eabc6216.
[29] W. Zheng, G. Dignon, M. Brown, Y. C. Kim and J. Mittal, *J. Phys. Chem. Lett.*, 2020, **11**, 3408–3415.
[30] G. L. Dignon, W. Zheng, Y. C. Kim, R. B. Best and J. Mittal, *PLoS Comput. Biol.*, 2018, **14**, e1005941.
[31] R. A. Patel, C. H. Borca and M. A. Webb, *Mol. Syst. Des. Eng.*, 2022, **7**, 661–676.
[32] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen *et al.*, *Comput. Phys. Commun.*, 2022, **271**, 108171.
[33] B. Rost and C. Sander, *Nature*, 1992, **360**, 540–540.
[34] B. Rost and C. Sander, *Proc. Natl. Acad. Sci.*, 1993, **90**, 7558–7562.
[35] B. Rost and C. Sander, *J. Mol. Biol.*, 1993, **232**, 584–599.
[36] N. Thapa, M. Chaudhari, S. McManus, K. Roy, R. H. Newman, H. Saigo and D. B. Kc, *BMC Bioinform.*, 2020, **21**, 1–10.
[37] M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. Von Lilienfeld, *Phys. Rev. Lett.*, 2012, **108**, 058301.
[38] N. E. Jackson, A. S. Bowen, L. W. Antony, M. A. Webb, V. Vishwanath and J. J. de Pablo, *Sci. Adv.*, 2019, **5**, eaav1190.
[39] R. K. Das and R. V. Pappu, *Proc. Natl. Acad. Sci.*, 2013, **110**, 13392–13397.
[40] A. Schlessinger, M. Punta and B. Rost, *Bioinformatics*, 2007, **23**, 2376–2384.
[41] T. Firman and K. Ghosh, *J. Chem. Phys.*, 2018, **148**, 123305.
[42] D. Sundaravadivelu Devarajan, S. Rekhi, A. Nikoubashman, Y. C. Kim, M. P. Howard and J. Mittal, *Macromolecules*, 2022, **55**, 8987–8997.
[43] R. Wuttke, H. Hofmann, D. Nettels, M. B. Borgia, J. Mittal, R. B. Best and B. Schuler, *Proc. Natl. Acad. Sci.*, 2014, **111**, 5213–5218.
[44] E. Van Dijk, P. Varilly, T. P. Knowles, D. Frenkel and S. Abeln, *Phys. Rev. Lett.*, 2016, **116**, 078101.
[45] K. A. Dill, D. O. Alonso and K. Hutchinson, *Biochemistry*, 1989, **28**, 5439–5449.
[46] G. L. Dignon, W. Zheng, Y. C. Kim and J. Mittal, *ACS Cent. Sci.*, 2019, **5**, 821–830.