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

Tzu-Hsuan Chao,[†] Shiv Rekhi,[‡] Jeetain Mittal,[‡] and Daniel P. Tabor*[,†]

†*Department of Chemistry, Texas A&M University, College Station, TX 77843 USA*

‡*Department of Chemical Engineering, Texas A&M University, College Station, TX 77843 USA*

E-mail: daniel_tabor@tamu.edu

Contents

S1 Link to Repository Containing Code and Raw Data

The raw data, code, and results for this paper can be found at the following Github repository: https://github.com/Tabor-Research-Group/Predicting-Disordered-Polymer-Folding-Behavior-Directly-from-Sequences

S2 Simulation Snapshots

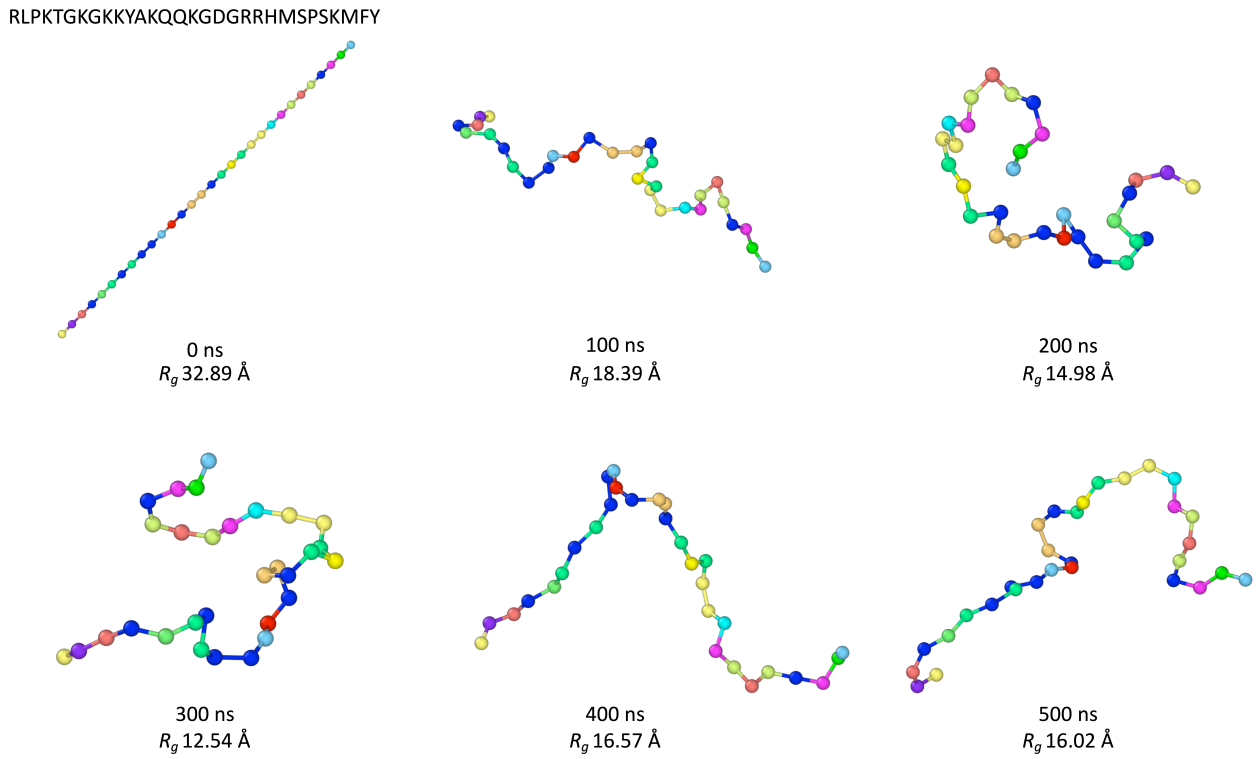


Figure S1

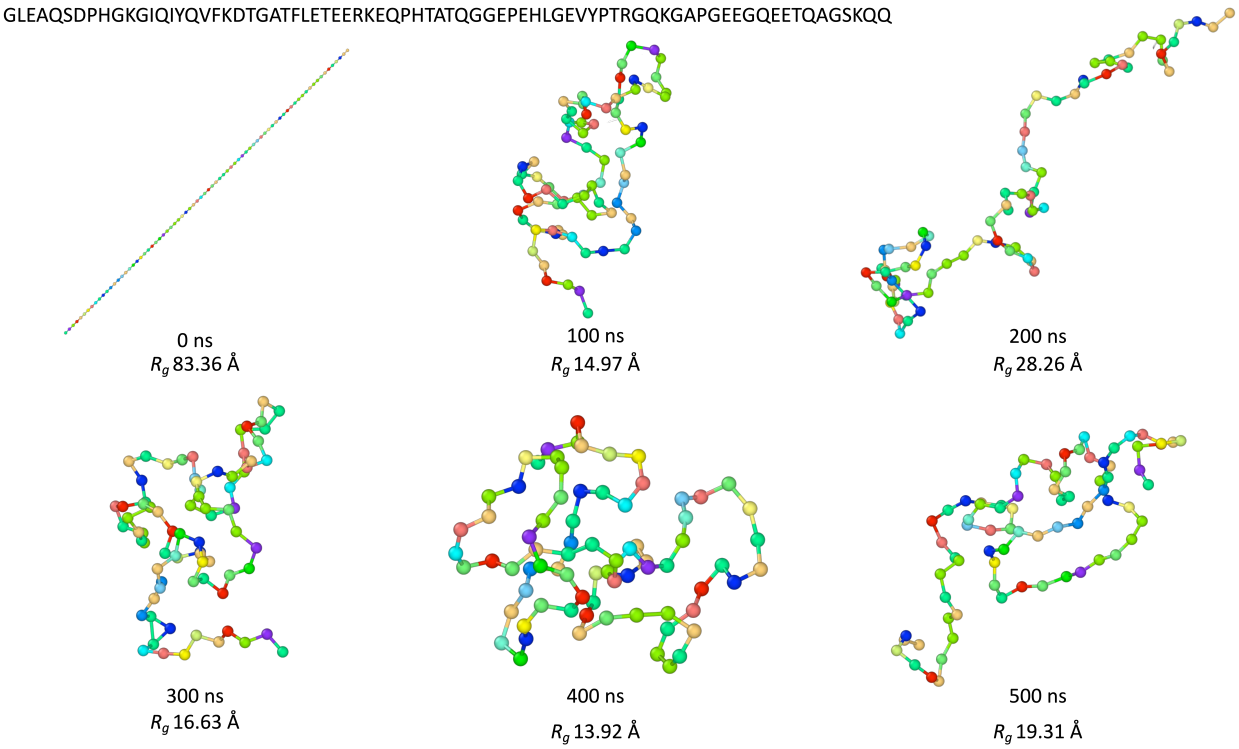

GLEAQSDPHGKGIQIYQVFKDTGATFLETEERKEQPHTATQGGEPEHLGEVYPTRGQKGAPGEEGQEETQAGSKQQ
0 ns
R_g 83.36 Å
100 ns
R_g 14.97 Å
200 ns
R_g 28.26 Å
300 ns
R_g 16.63 Å
400 ns
R_g 13.92 Å
500 ns
R_g 19.31 Å

Figure S2

SYGAMIKESDFEEDAQEDEIAERCNLYISAPLEIAVTFLAGYSTKHGTLLLALYKIPGSVAAKWVEVGKNMASLYPAIVHIVVVFHRLSADGLPQYQGHVGQKAPA
VVGLARPVQEVAASCLYAQDRIGLSLKNVLHMEAGLPSAKGFKSEGNKTVQLPVAKNMPAGKPFGAPVASAKEEEVSFLPFAGA

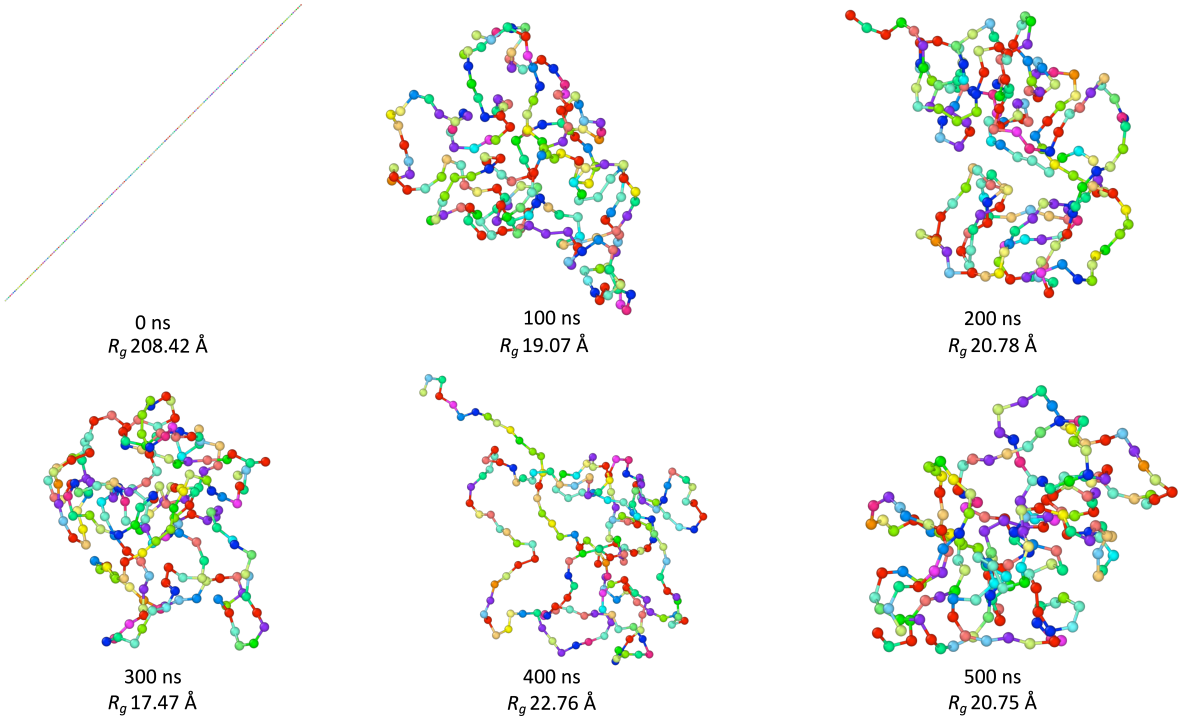


Figure S3

S3 Illustrating Each Encoding Method on One Example Sequence

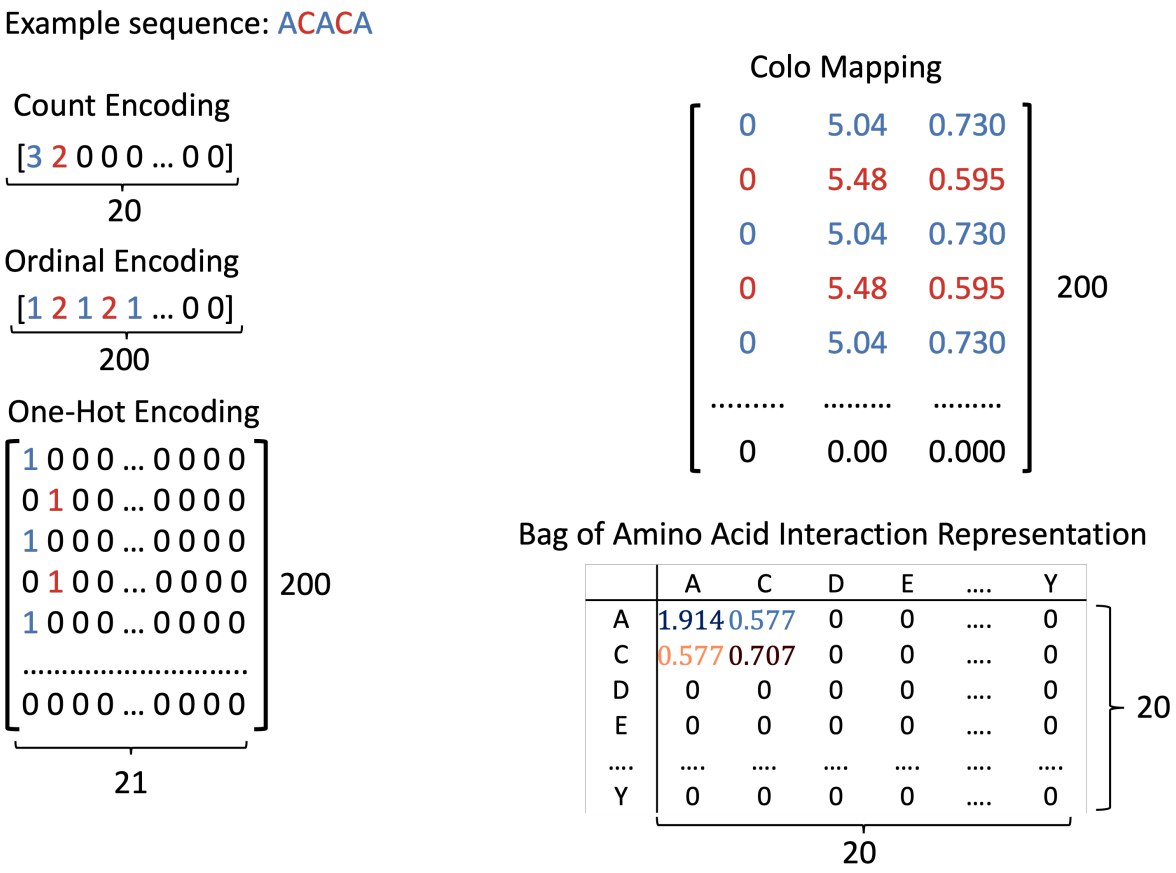
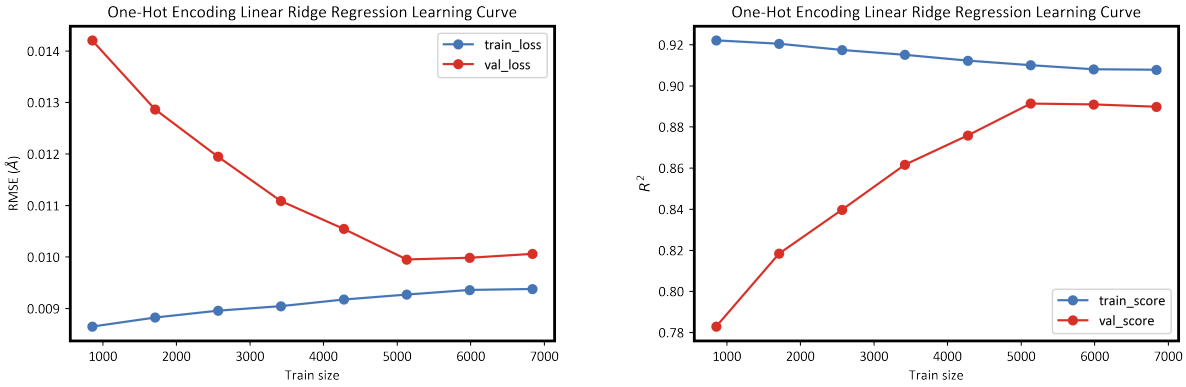


Figure S4: Five encoding methods: categorical featurization (left) and physically-motivated featurization (right).

S4 One-Hot Encoding Linear Regression Learning Curves

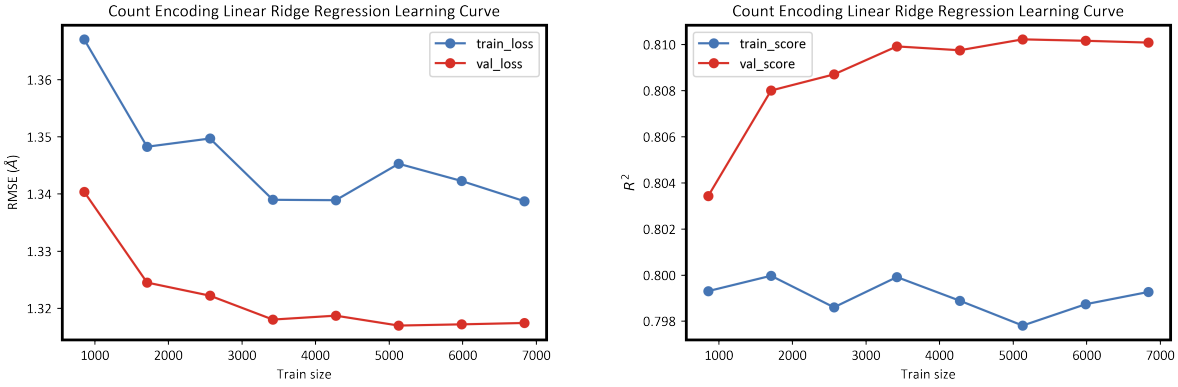
Figures S5a and S5b contain the learning curves for the one-hot encoding linear regression models. The learning curve of one-hot encoding indicates that the model overfits for these sparse features, as the validation loss increases at the last three points as the training set size increases.



(a) Loss (b) Score

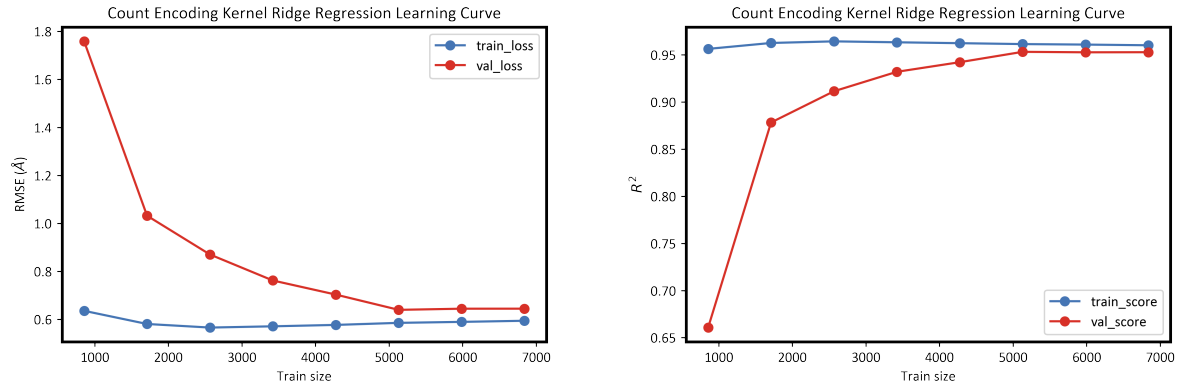
Figure S5: One-Hot Encoding Linear Regression Learning Curve

S5 Learning Curves



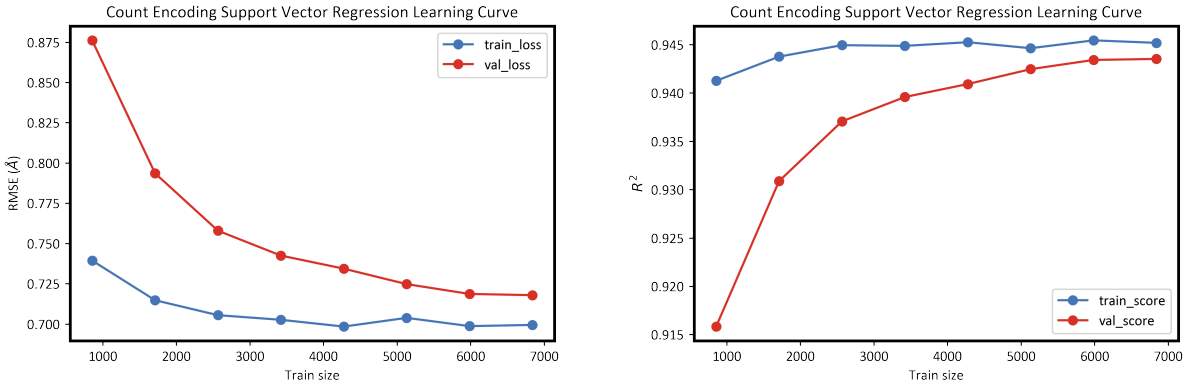
(a) Loss (b) Score

Figure S6: Count Encoding Linear Ridge Regression Learning Curve



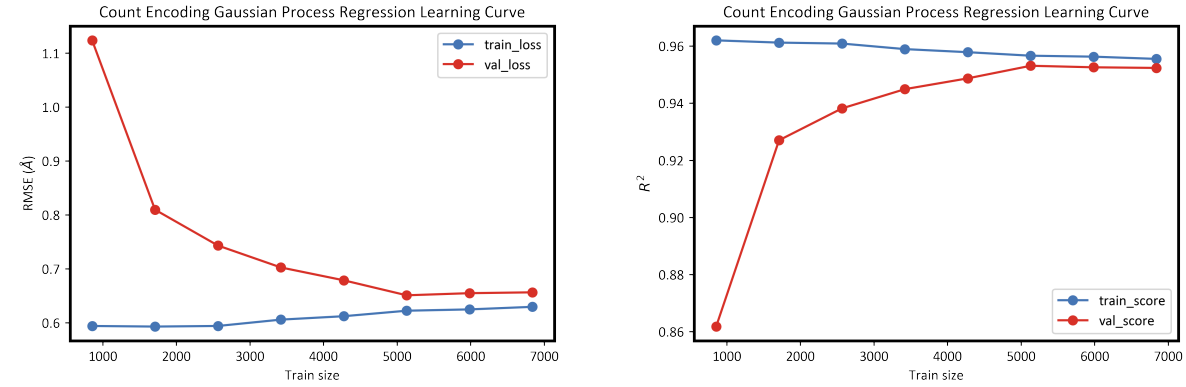
(a) Loss (b) Score

Figure S7: Count Encoding Kernel Ridge Regression Learning Curve



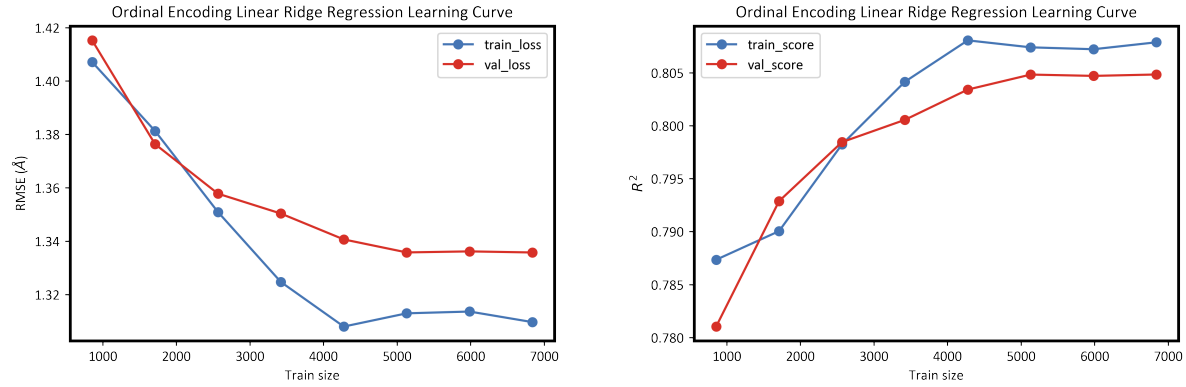
(a) Loss (b) Score

Figure S8: Count Encoding Support Vector Regression Learning Curve



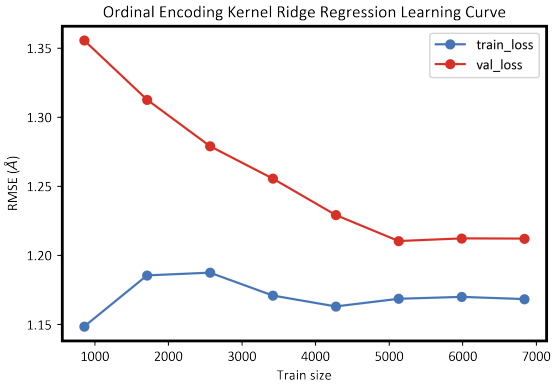
(a) Loss (b) Score

Figure S9: Count Encoding Gaussian Process Regression Learning Curve

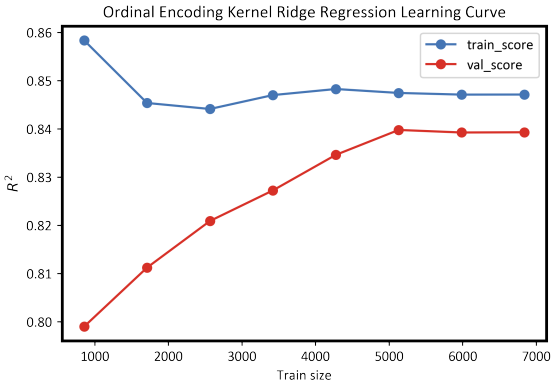


(a) Loss (b) Score

Figure S10: Ordinal Encoding Linear Ridge Regression Learning Curve

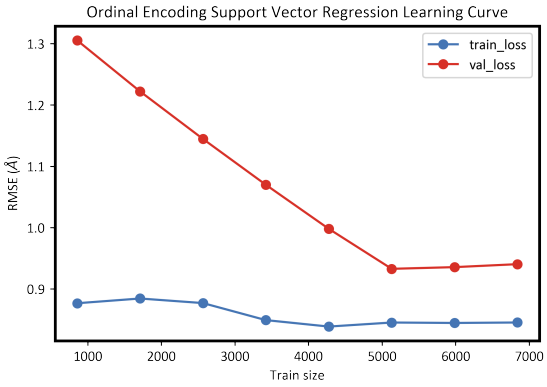


(a) Loss

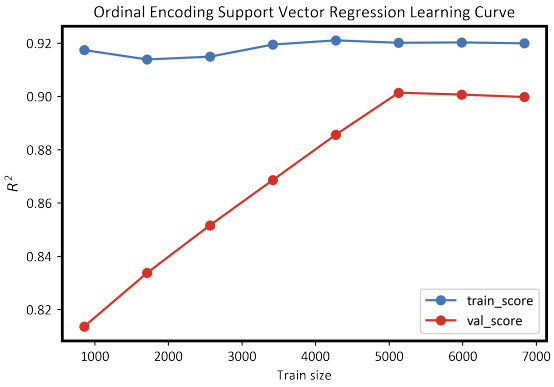


(b) Score

Figure S11: Ordinal Encoding Kernel Ridge Regression Learning Curve

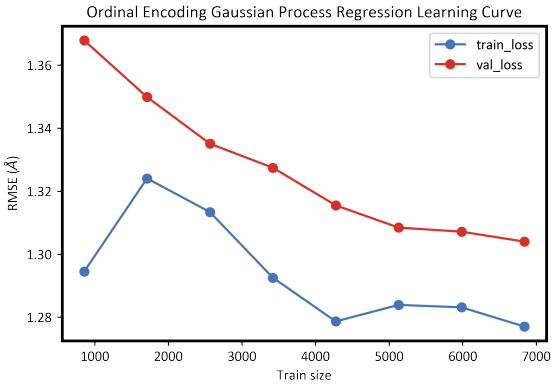


(a) Loss

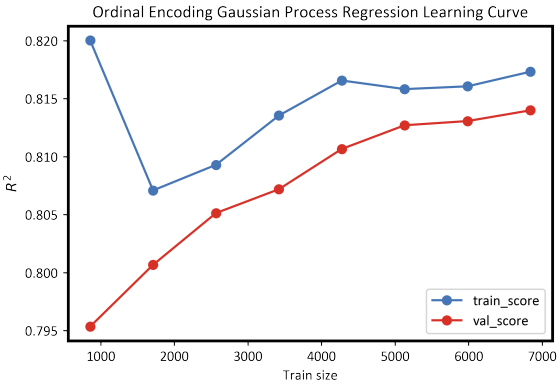


(b) Score

Figure S12: Ordinal Encoding Support Vector Regression Learning Curve

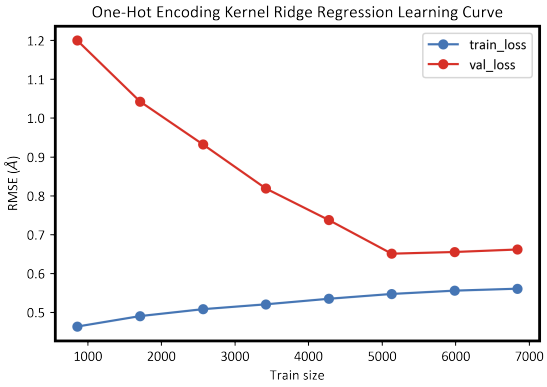


(a) Loss

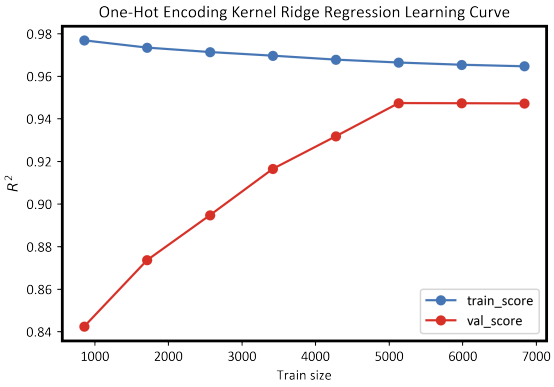


(b) Score

Figure S13: Ordinal Encoding Gaussian Process Regression Learning Curve

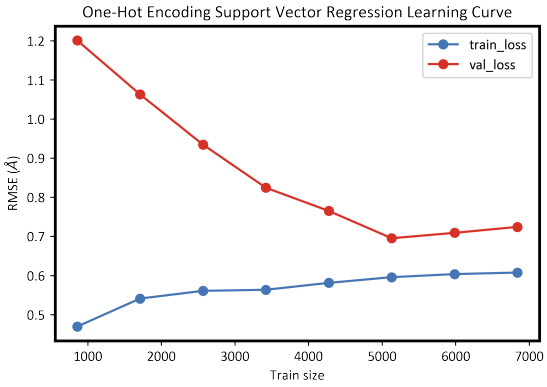


(a) Loss

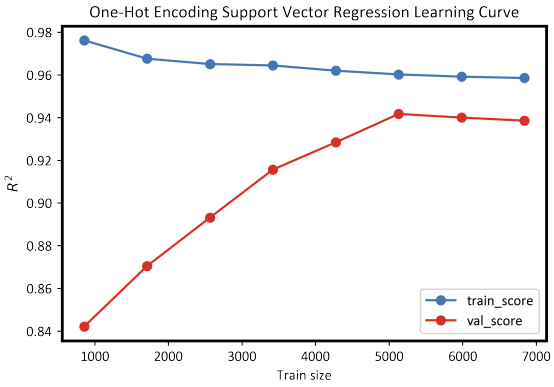


(b) Score

Figure S14: One-Hot Encoding Kernel Ridge Regression Learning Curve

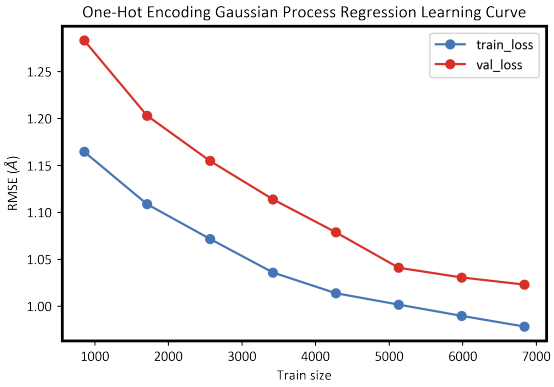


(a) Loss

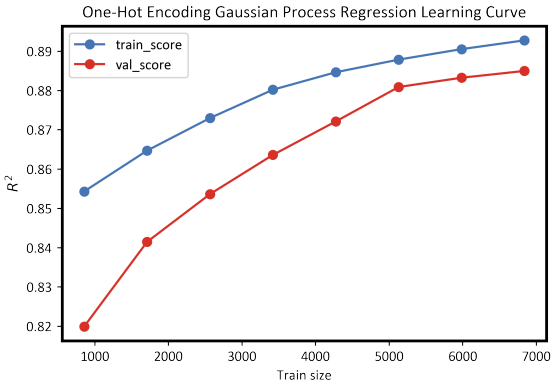


(b) Score

Figure S15: One-Hot Encoding Support Vector Regression Learning Curve



(a) Loss



(b) Score

Figure S16: One-Hot Encoding Gaussian Process Regression Learning Curve

S6 2D Convolutional Neural Network

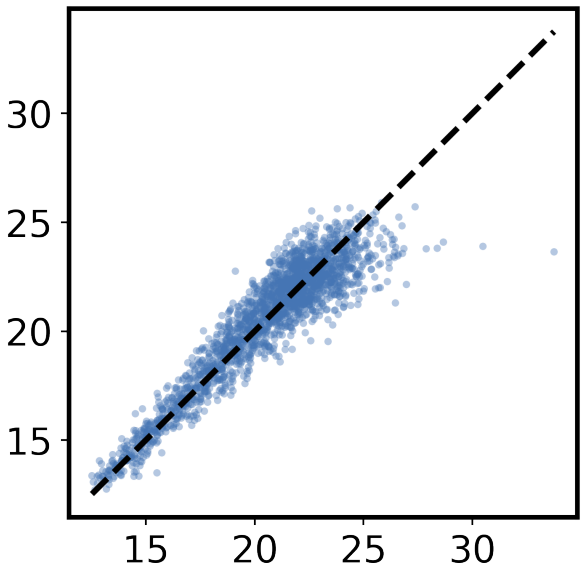


Figure S17: Prediction using the Color Mapping representation and a2D convolutional neural network model.

S7 Bag of Amino Acid Representation: Testing the Exponent in the Representation

Here, we tested the performance of a family of BAAI featurizations. The exponent that is used for each matrix element in the feature (β):

$$BAAI = \sum_{i-j>1} (X_i - X_j)^{\beta} \tag{1}$$

was varied, similar to the approach employed in reference S1. For each of these formulations of the feature, we conducted the training and testing procedure as the model used in the main text. For these tests, we used support vector regression, as it was the best model for the baseline model. The first plot is the testing performance and the second plot is the extrapolation test using the same strategy as the main text but different exponent values used in each element.

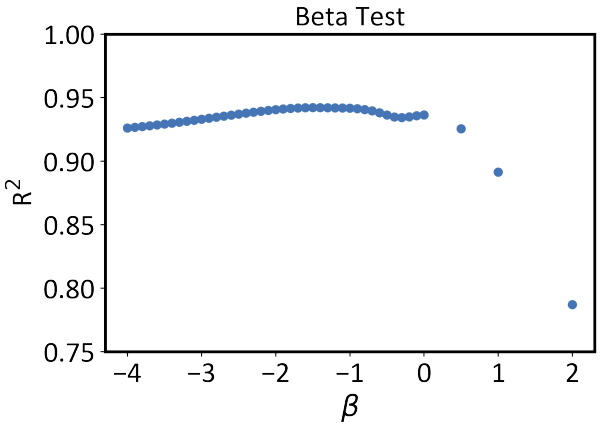


Figure S18

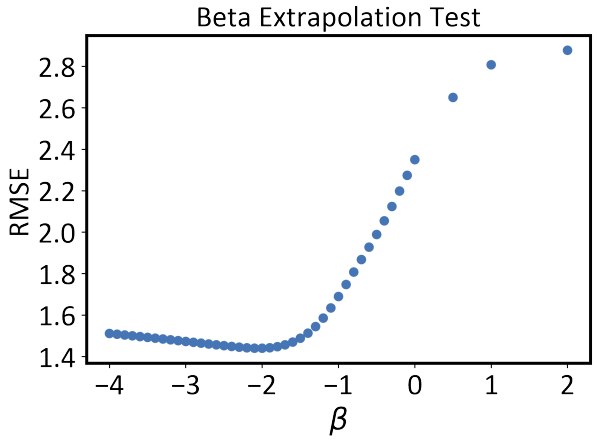


Figure S19

S8 Count Encoding Extrapolation Test

We performed extrapolation tests (as described in section 3.4 of the main text) on count encoding. For this feature, we only tested the extrapolation performance with a support vector regression model, as the count encoding only contains 20 features. The performance is similar to BAAI model on the IDP-10260 dataset.

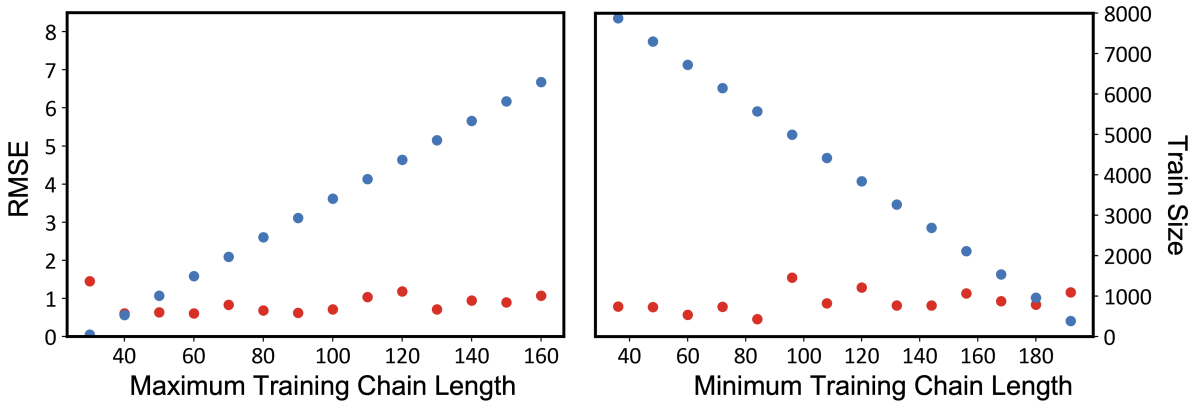
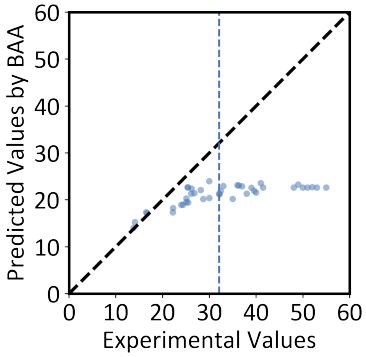


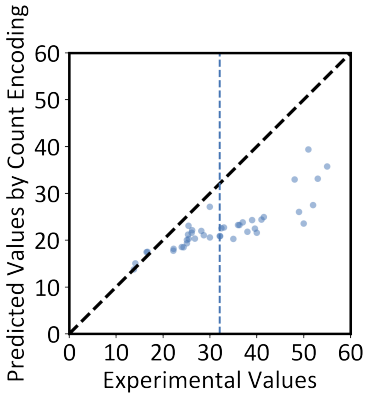
Figure S20

S9 Prediction on Experimental Data

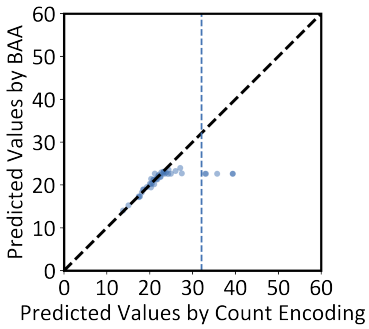
We apply both CE and BAAI models on 42 experimental data points. Three plots of correlation between prediction and experimental values are provided.



(a) Comparison between experimental values and prediction from BAAI model



(b) Comparison between experimental values and prediction from CE model



(c) Comparison between prediction from CE and BAAI model

Figure S21: Prediction on Experimental Data

S10 Reverse Sequences Testing

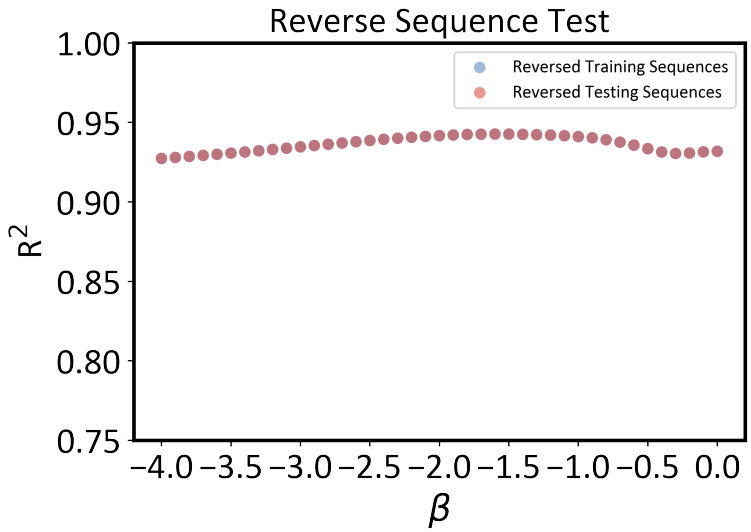


Figure S22

As the underlying coarse-grained simulations do not depend on the amino acid sequence, we can test the performance of the trained models on predicting the properties of "reversed sequences." By virtue of its construction, the count encoding feature will have the same predictions, but the directional BAAI representation will have different representations for reversed sequences.

We tried 1) only reversing the training sequences or 2) only reversing the testing sequences. The resulting R^2 (as a function of β) shows that both cases give similar performance. This indicates that the BAAI feature can handle data that has properties that are symmetric upon reversal of the sequence while the generated matrix is transposed. This test was to ensure that this model, with this representation, doesn't "over-learn" from the intrinsically disordered proteins.

S11 Symmetric BAAI

Since the coarse-grained model that was used to generate the training data is "symmetric" (it would give identical results if the sequence were reversed), we built a symmetric representation, which encodes both forward (*e.g.,* $C{\rightarrow}A$) and reverse (*e.g.,* C$\leftarrow$A) interactions together in a single interaction. To make a non-redundant version of symmetric BAAI representation, we tested training the model with only 210 features (only the "upper triangle" of the BAAI matrix). The testing performance is similar to the original representation.

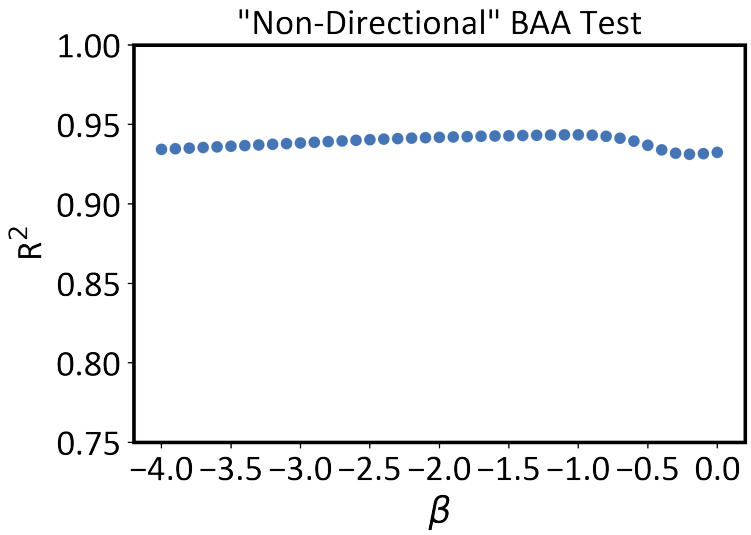


Figure S23: Performance (R^2 of the test set) of the symmetric BAAI representation (where are all interactions are symmetric, instead of proceeding along the chain) for the SVR models as a function of β.

S12 Shuffled Test

Here, we shuffle the R_g of a specific portion of training data and test using the correct R_g. The R^2 values between shuffled and non-shuffled training data are plotted to see how robust the representation is to training set errors, which might be seen in larger, future datasets. The testing performance can still achieve over 0.75 R^2 when 40% of the training data is shuffled.

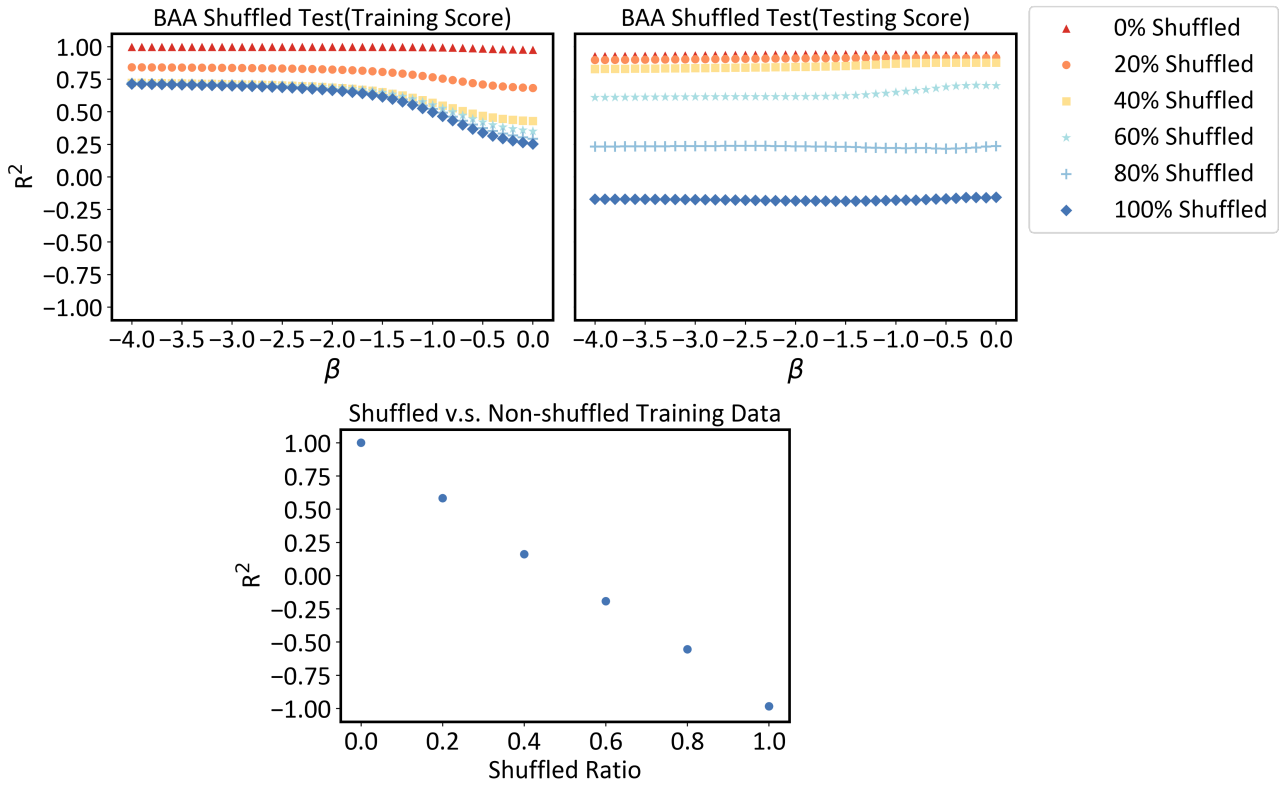


Figure S24: Performance of training and test set scores for BAAI representation (with SVR model) when the model is given "erroneous" random sequence data as a fraction of its training set.

S13 Shuffle Test-Count Encoding

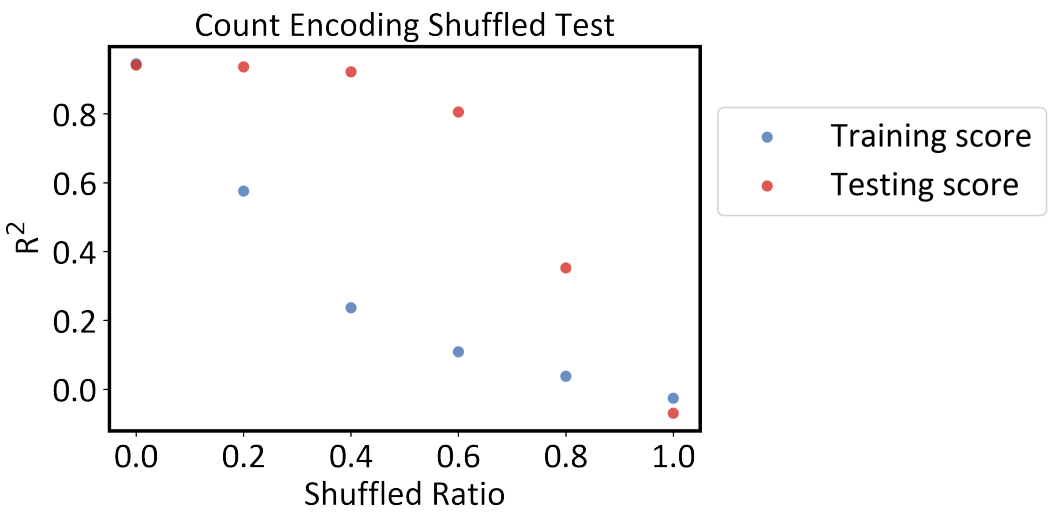


Figure S25

The shuffle test is also performed on count encoding. The training score goes down (as it should), while the testing score can still maintain an R^2 of 0.8 when 60% of the data is shuffled.

S14 Simulation Details

All the simulations were performed using LAMMPS. A Langevin thermostat with a friction coefficient of 1 ps^{-1} was used to for each simulation. Each simulation consisted of a 500 ns run. The first 100 ns was used for the system to reach equilibrium. Seven temperatures (270, 300, 330, 360, 390, 420, 450K) and 10 fs timestep were used for all sequences. Bonded, electrostatic and short-range pairwise interactions defined by amino acid hydropathy are included in the simulations.[S2] Bonded interactions are modeled using a harmonic potential with a spring constant of 10 kcal/mol $Å^2$ and a bond length of 3.8 Å. Electrostatic interactions are described using Debye-Huckel electrostatic screening:

$$E_{ij}(r) = \frac{q_i q_j}{4\pi\epsilon_0 D r} \exp\left(-r/\kappa\right) \tag{2}$$

where κ is the Debye screening length and D is the dielectric constant of the solvent, 80 for water. The short-range pairwise interaction is described by Ashbaugh-Hatch functional form,

$$\Phi(r) = \begin{cases} \Phi_{LJ} + (1-\lambda)\epsilon, & \text{if r} \leq 2^{\frac{1}{6}}\sigma \\ \lambda\Phi_{LJ}, & \text{otherwise} \end{cases} \tag{3}$$

where Φ_{LJ} is the standard Lennard-Jones potential. The lambda value is calculated by the average of the hydropathy values of two amino acids (Table S1).

The scaling exponent ν is derived using the formulation of Zheng *et al.*[S1]

$$R_g = \sqrt{\frac{\gamma\left(\gamma+1\right)}{2\left(\gamma+2\nu\right)\left(\gamma+2\nu+1\right)}} b N^{\nu} \tag{4}$$

where $\gamma = 1.1615$, $b = 0.55$ and N is the number of peptide bonds.

S15 Coarse-grained Model Parameters

Table S1: CG model parameters

Amino Acid	Charge	σ(Å)	λ
A	0	5.04	0.730
C	0	5.48	0.595
D	-1	5.58	0.378
E	-1	5.92	0.459
F	0	6.36	1.000
G	0	4.50	0.649
H	0.5	6.08	0.514
I	0	6.18	0.973
K	1	6.36	0.514
L	0	6.18	0.973
M	0	6.18	0.838
N	0	5.68	0.432
P	0	5.56	1.000
Q	0	6.02	0.514
R	1	6.56	0.000
S	0	5.18	0.595
T	0	5.62	0.676
V	0	5.86	0.892
W	0	6.78	0.946
Y	0	6.46	0.865

S16 Best Parameters for Classical Regression Models

Count Encoding:

Linear Ridge Regression: alpha: 0.6

Kernel Ridge Regression: alpha: 0.1, gamma: 1

Support Vector Regression: C: 100, gamma: 0.1 epsilon; 0.1

Gaussian Process Regression: alpha:0.05, length scale:1e5, length bounds:1e-20 $\sim$ 1e20

Ordinal Encoding:

Linear Ridge Regression: alpha: 350

Kernel Ridge Regression: alpha: 0.6, gamma: 0.01

Support Vector Regression: C: 1, gamma: 0.1 epsilon; 0.1

Gaussian Process Regression: alpha:0.7, length scale:1e5, length bounds:1e-20$\sim$ 1e20

One-Hot Encoding:

Linear Ridge Regression: alpha: 120

Kernel Ridge Regression: alpha: 0.1, gamma: 0.001

Support Vector Regression: C: 10, gamma: 0.001 epsilon; 0.001

Gaussian Process Regression: alpha:1.5, length scale:1e5, length bounds:1e-20$\sim$ 1e20

BAAI representation:

Support Vector Regression: C:10 gamma:0.1 epsilon:0.1

S17 Artificial Neural Network Architecture

Feed Forward Neural Network(section 3.5)

Input layer: dimension:400, activation function: swish, kernel initializer: glorot uniform

One hidden layer: dimension:400, activation function: swish, kernel initializer: glorot uniform

Output layer: dimension:2, activation function: linear, kernel initializer: glorot uniform

Conv1D neural network(section 3.2)

One conv1d layer: filter:8, kernel size:3, stride:1, activation function: relu, kernel initializer: lecun normal

One hidden layer: dimension:10, no activation function, kernel initializer: lecun normal, dropout rate:0.2

output layer: dimension:1

Conv2D neural network(section 3.2)

One conv2d layer: filter:8, kernel size:(3,3) , stride:(1,1) , activation function: relu, kernel initializer: lecun normal

One hidden layer: dimension:10, no activation function, kernel initializer: lecun normal, dropout rate:0.2

output layer: dimension:1

S18 Comparison of BAAI Representation to Augmented Fingerprints and Performance on Alternative IDP Datasets

Here, we test the performance of a set of the BAAI representations (specific details indicated in the figures) on the set of 2585 intrinsically disordered proteins given in Reference S3. We compare our results on both datasets to the augmented fingerprint approach described in the same reference, on both the randomly generated 10260 dataset in this paper and the 2585 dataset. The overall results are summarized in Table S2, with details shown in the text and figures that follow.

Table S2: Test set R^2 for BAAI representations and Augmented Fingerprint Representations (SVR Models)

	Count Encoding	**Augmented Fingerprint**	**BAAI**	**BAAI + CL**	**Symmetric BAAI**	**Symmetric BAAI + CL**	
IDP-10260 (This work)	0.942	0.961	0.936	0.937	0.937	0.937	
IDP-2585 set [S3]	0.964	0.983	0.953	0.965	0.948	0.969	(β=-0.5)
			0.967	0.967	0.974	0.974	(β=-2.0)

Although the BAAI + SVR model showed some ability to extrapolate to longer sequence lengths in the 10260 dataset, the 2585 dataset contains much longer sequences (similar in length to the sequences shown in Figure S21). In addition, the 2585 simulation dataset was obtained with the temperature-corrected coarse-grained model, whereas we used a model where the potential is not temperature-dependent for the 10260 dataset. Similar to our results in Section S9, this means that the model was trained on a just the IDP-10260 set does not extrapolate well to higher values of R_g. However, we are able to improve the BAAI + SVR model with retraining, which we show both in Table S2 above and the figures below.

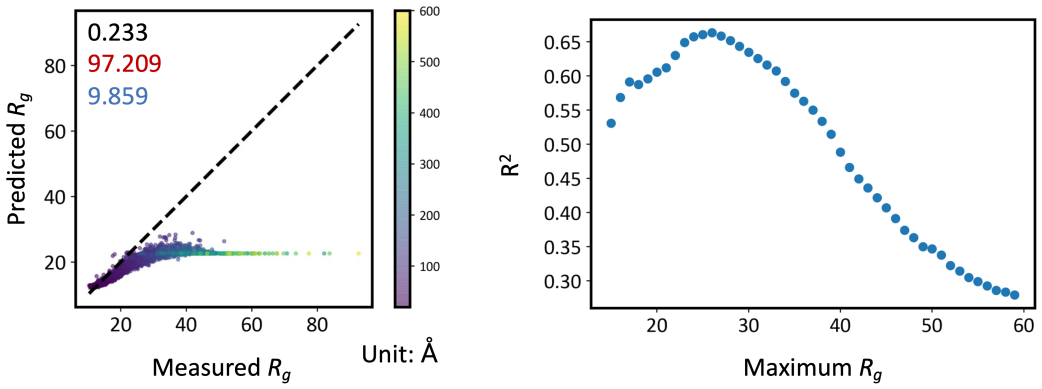


Figure S26: Direct prediction or R_g based on BAAI representation (SVR model) trained on our data set. The R^2, MSE, and RMSE are indicated on the inset of the plot, indicating that retraining is necessary.

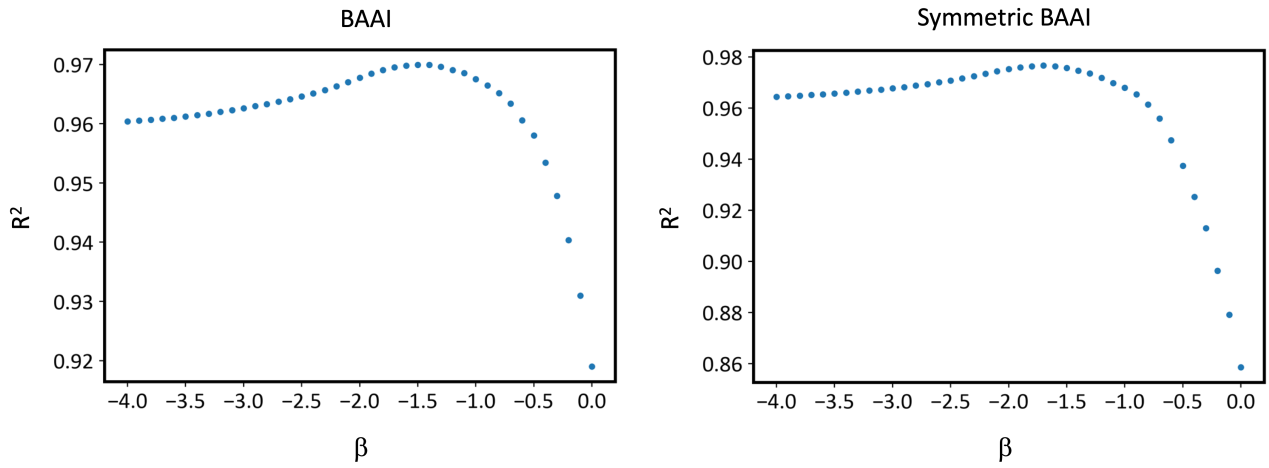


Figure S27: Retraining on the IDP-2585 set. A test on the exponent β was performed to evaluate the sensitivity of the model.

Though the BAAI models do not reach the same accuracy as the augmented fingerprint models, we expect that some of this behavior could be explained by the size of the dataset. Since all BAAI representations have at least 210 features, they may need a larger volume of training data to show a payoff in improved performance, though the performance is relatively good, especially for the symmetric BAAI with $\beta = -2.0$, where only a few points could make difference in the model's top-line R^2 performance. In addition, as mentioned in the main text, certain sequences may need to be explicitly included to challenge the models more than randomly generated (or DisProt-sourced) sequences.

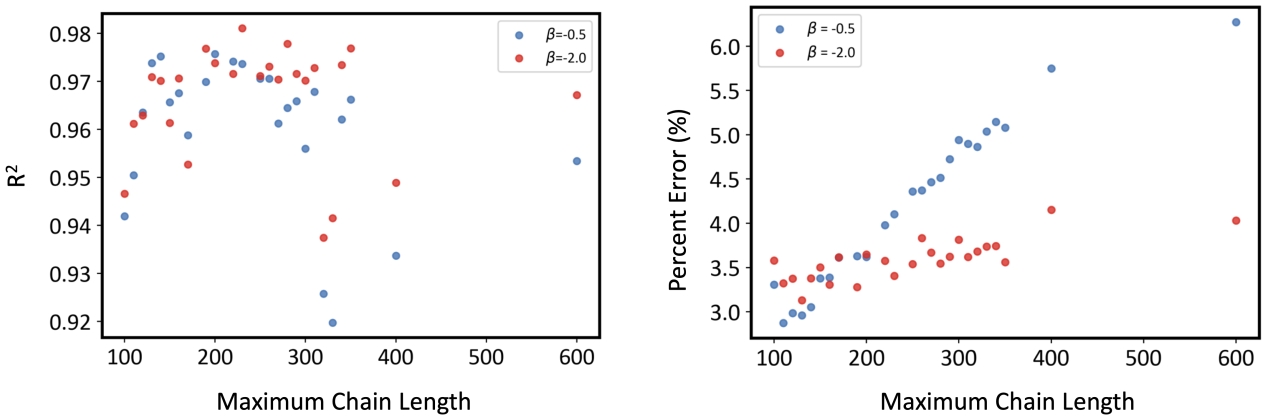


Figure S28: Extrapolation test using different β values in BAAI on the 2585 dataset. The R^2 values and percent error are plotted to examine the extrapolation ability of β values of -0.5 or -2.0.

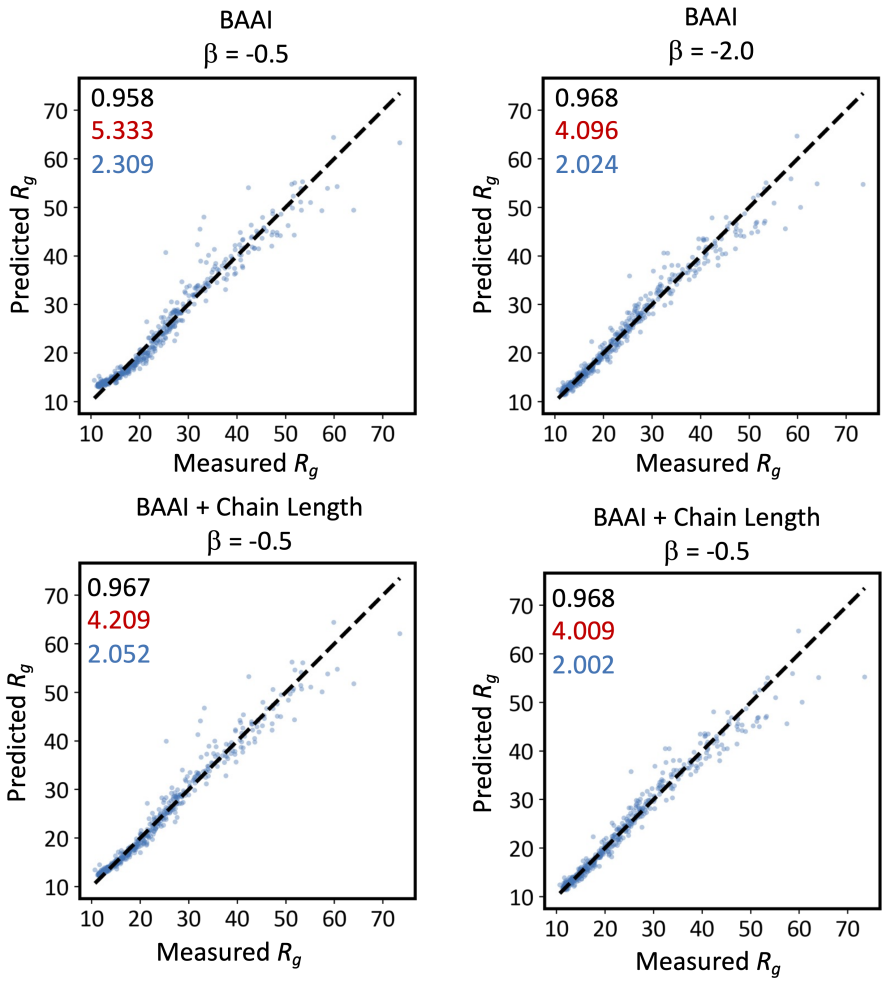


Figure S29: Performance of BAAI and BAAI + Chain Length using different β values.

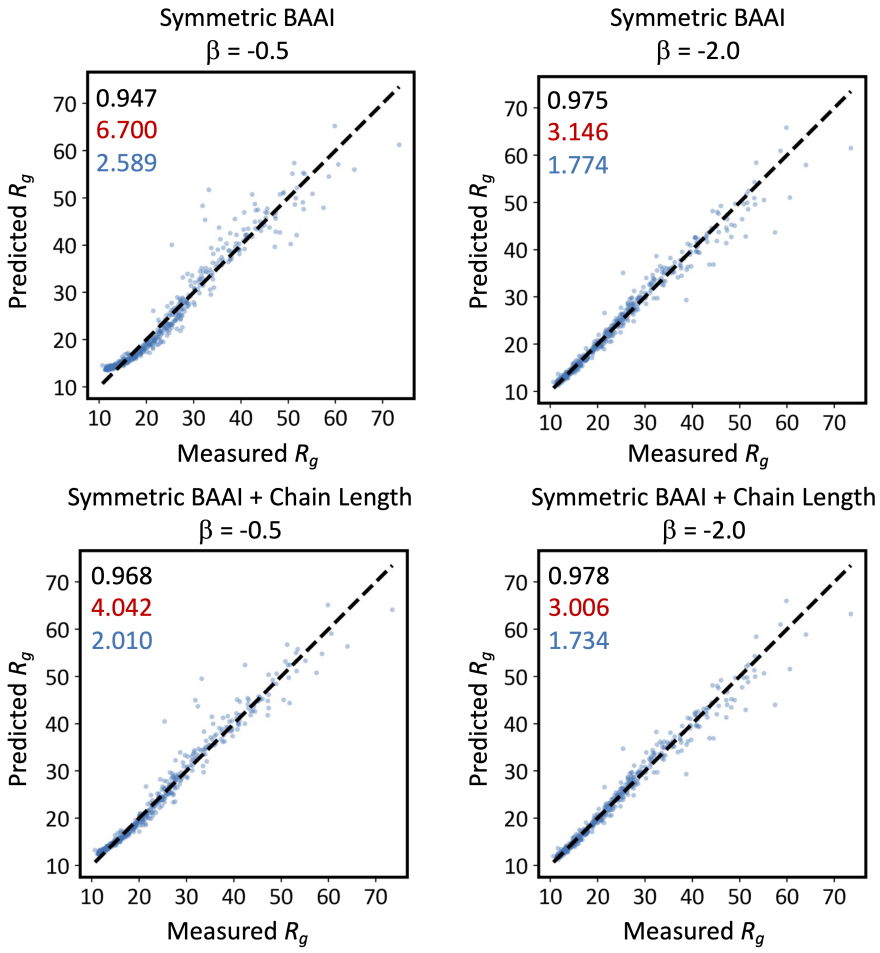


Figure S30: Performance of symmetric BAAI and symmetric BAAI + Chain Length (CL) using different β values.

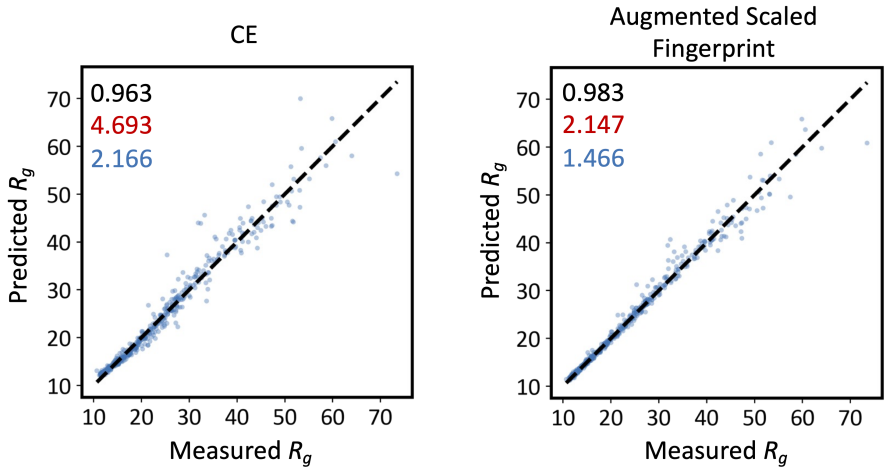


Figure S31: Performance of count encoding and augmented fingerprint

S19 Analysis of Temperature-Dependent Model for Temperature Subsets

The following figures show the results for the models presented in Figure 5 of the main text but broken down by performance in each temperature. We present the data in four different ways due to the changing and length scales for the R_g of the polymer as a function of temperature. First, we present the ν and R_g scatter plots as shown in Figure 5 in the text. Then, we show the R^2, MAE, RMSE, and average percent error predictions as a function of temperature for both of these metrics. Overall, the models are relatively consistent across different temperatures. However, because R_g increases as a function of temperature, MAE and RMSE will increase in parallel. In the intermediate temperatures, we also see an increase in the MAE and ν (which, although it should be length-independent, has previously been noted to be length-dependent for this dataset in the text). Finally, as all of the IDPs become, on average more "random walk" like at higher temperatures (note the clustering of the data above 360 K on ν in Figure S32), it becomes more difficult to differentiate the polymers from each other, and thus, R^2 decreases, even though the percent error in the prediction remains the same.

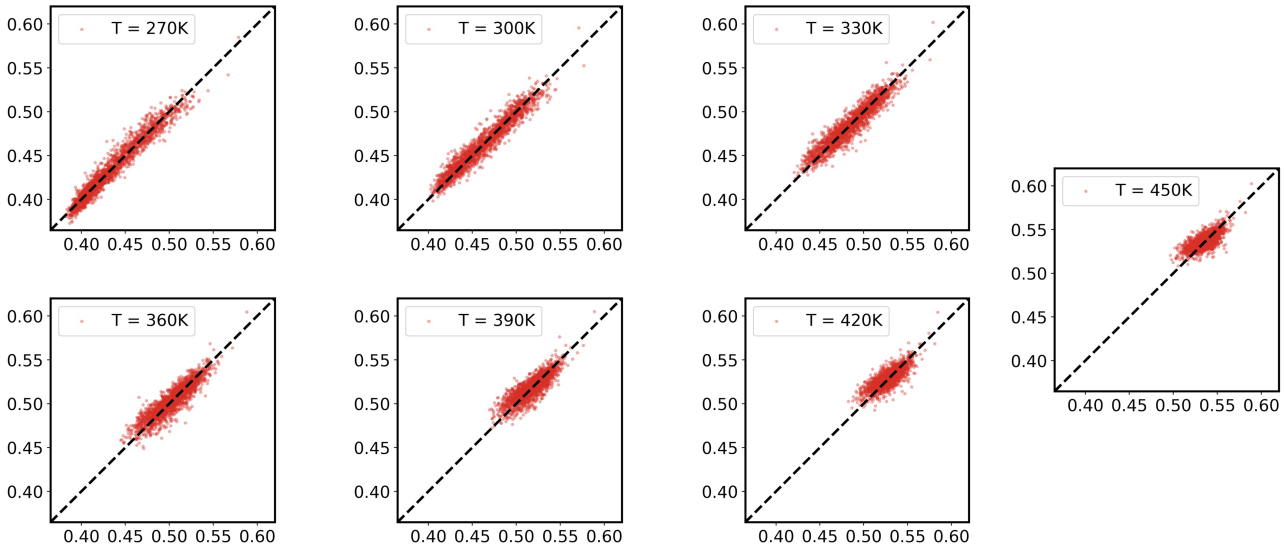


Figure S32: Performance of temperature-incorporated model for each temperature for ν.

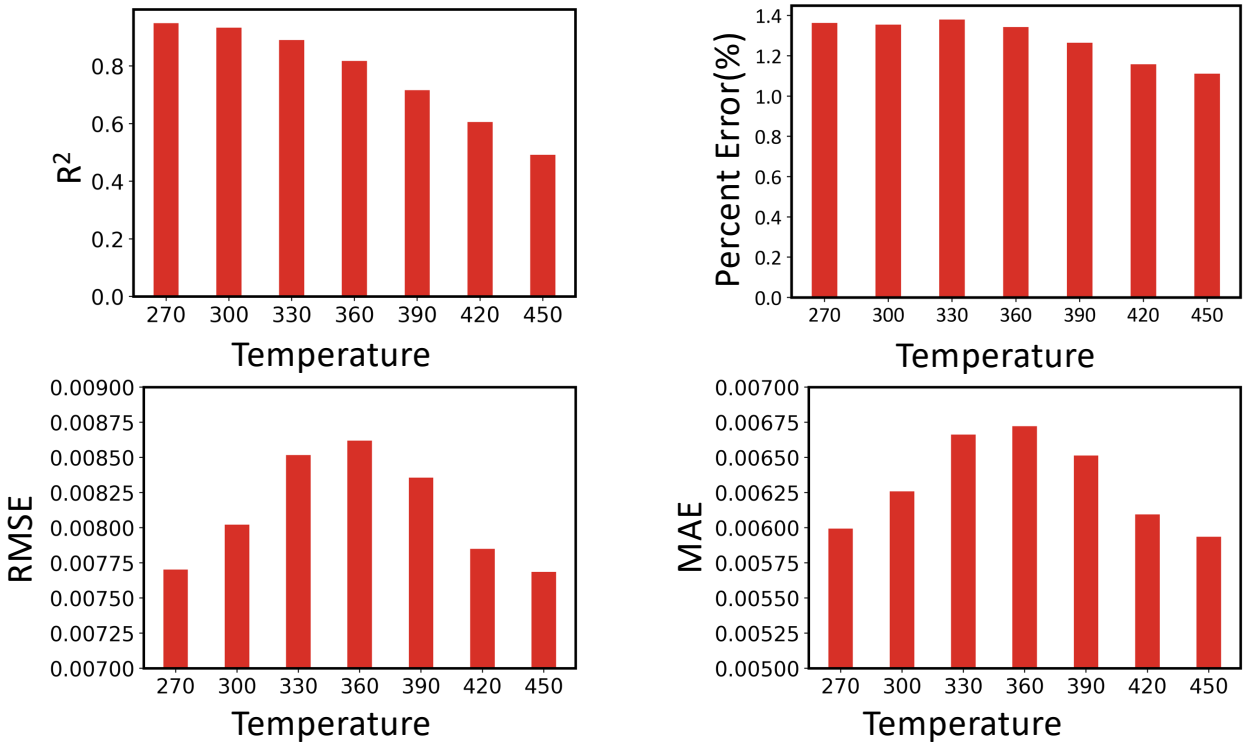


Figure S33: Performance of temperature-incorporated model for each temperature for ν.

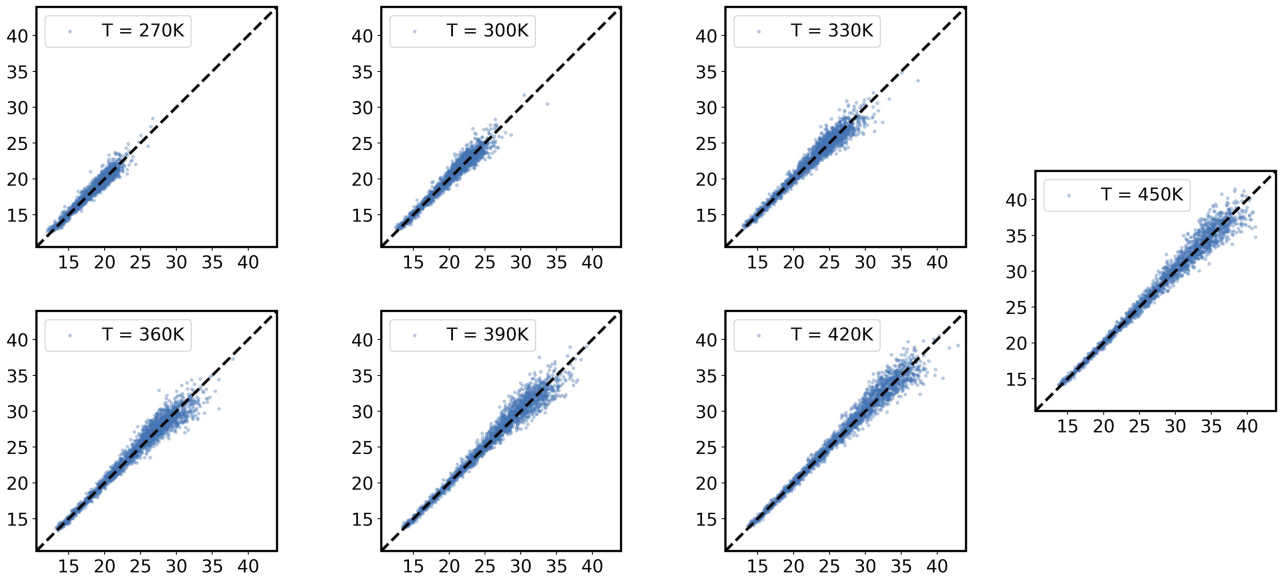


Figure S34: Performance of temperature-incorporated model for each temperature for R_g

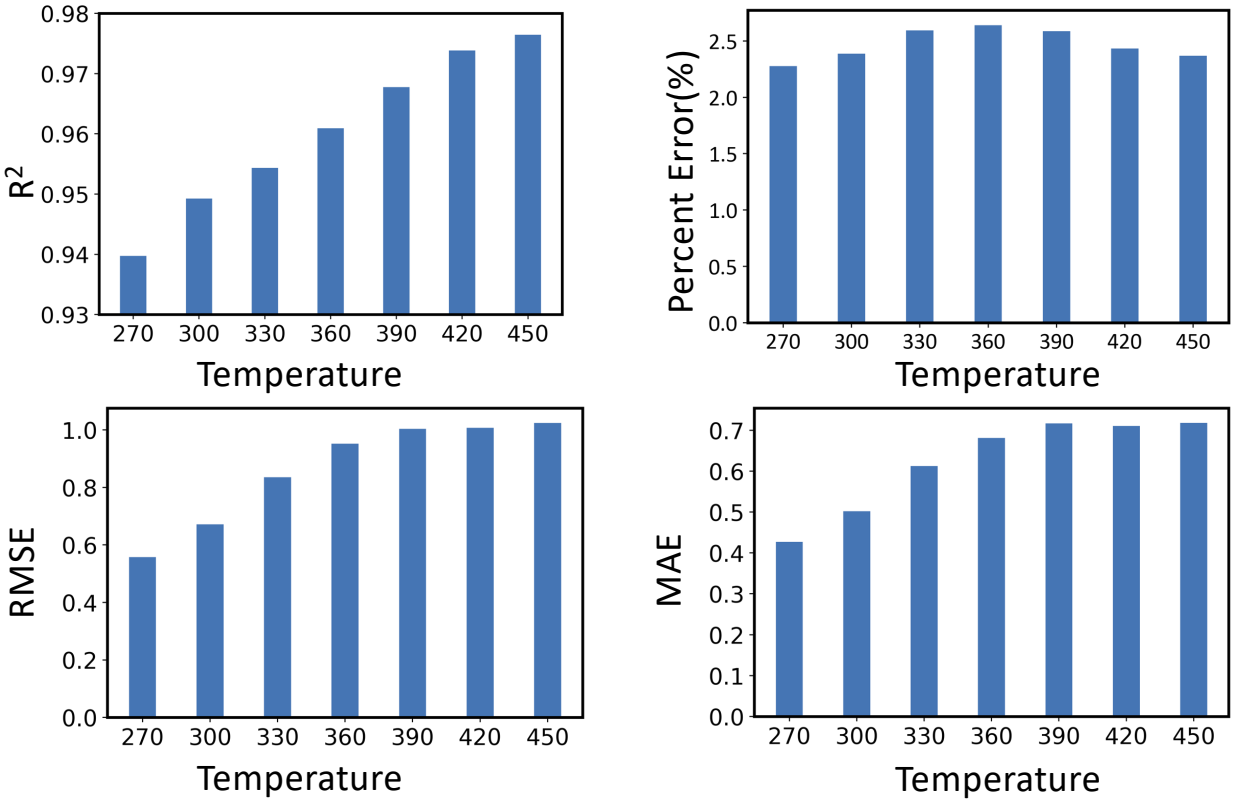


Figure S35: Performance of temperature-incorporated model for each temperature for R_g

References

[S1] Zheng, W.; Dignon, G.; Brown, M.; Kim, Y. C.; Mittal, J. Hydropathy patterning complements charge patterning to describe conformational preferences of disordered proteins. *J. Phys. Chem. Lett.* **2020**, *11*, 3408–3415.

[S2] Dignon, G. L.; Zheng, W.; Kim, Y. C.; Best, R. B.; Mittal, J. Sequence determinants of protein phase behavior from a coarse-grained model. *PLOS Comput. Biol.* **2018**, *14*, e1005941.

[S3] Patel, R. A.; Borca, C. H.; Webb, M. A. Featurization strategies for polymer sequence or composition design by machine learning. *Mol. Syst. Des. Eng.* **2022**, *7*, 661–676.