# **Supporting Information**

#### Machine Learning Approach for Accurate Backmapping of Coarse-Grained Models to All-

#### **Atom Models**

Yaxin An and Sanket A. Deshmukh\*

Department of Chemical Engineering, Virginia Tech, Blacksburg, VA 24061, USA

#### **S1. Dataset Construction**

The dataset to train the ML models consisted of coordinates of beads in CG models as input, and the coordinates of atoms in all-atom models as output. The accuracy and robustness of ML models rely on the fidelity of the dataset. Specifically, the training dataset should cover a large range of molecular structures.<sup>1,2</sup> To generate the input data for training and testing of all the ML models, 1000 molecules were randomly packed in a simulation box and then equilibrated for 5 ns in NPT ensemble. All-atom MD simulations were carried out with CHARMM force-field by using the NAMD package.<sup>3,4</sup> Periodicity was applied in all the three directions. A real space cutoff of 12 Å was used to truncate the nonbonded interactions with a switching function applied at 9 Å to truncate the van der Waals potential energy smoothly at the cutoff distance. Long-range Coulombic interactions were treated using particle mesh Ewald with an accuracy of  $1 \times 10^{-6}$ . A pair list distance of 15 Å was used to store the neighbors of a given bead. The equations of motion were integrated by the velocity Verlet algorithm with a timestep of 1 fs. Langevin thermostat and barostat were used to keep the temperature at 300 K and pressure at 1 bar. The positions of atoms were saved every 1 ps. We extracted a trajectory of 100 randomly selected molecules in the final 10 ps (100 frames) of 5 ns (50,000 frames), which contained 10,000 configurations in total for generating the dataset. The center of mass of every molecule from these 10,000 configurations was placed to the origin, which can be treated as normalizing the coordinates of each atom to a narrow range. This resulted in a small dimensional space for the dataset for training, to help improve the accuracy of ML models. An example of the dataset for hexane is shown in Figure S1. Two files of the dataset: data\_benzene.xlsx and data\_hexane.xlsx have been attached as supporting information for readers who are interested in training/improving the models in the future.

9-dimensional feature space containing cartesian coordinates of three CG beads				60-dimensional output space containing cartesian coordinates of 20 atoms										
x1	x2	x3		x7	x8	x9	y1	у2	у3		y57	y58	y59	y60
ł	1	1	1	1	1	1	1	1	1	1	ł	1	l	1

**Figure S1.** The dataset of hexane for building ML models. The coordinates of CG beads are in blue and those of atoms are in green. Similar datasets were used for other molecules in the present study.

Several methods such as k-fold, leave-one-out, etc. can be used to generate/split a dataset to train the ML models.<sup>2,5</sup> As one would expect each method has its own advantage and disadvantage so care must be taken while choosing these methods. Each dataset was randomly split so that 80 % of the data for training and the remaining 20 % for testing. For training, *k*-fold cross-validation was used to prevent overfitting of the ML models.<sup>2</sup> As shown in **Figure S2**, the training data is split k folds (k = 5 in this study), where one fold is used as a validation set and the other *k*-1 folds as training sets. This process is repeated *k* times with each fold as a validation set and thus *k* ML models were generated. The performance of these *k* ML models was averaged to get the final average R<sup>2</sup> score and standard deviation. This method is generally robust and yields results with reasonable accuracy, as demonstrated in the present study.



Figure S2. The schematic for k-fold (k=5) cross validation.

### S2. ML Model development

ANN is one of the most popular ML models, which has been widely used in information technologies such as image classification, and natural language processing.<sup>6</sup> Recently, it has also been used in computational materials science.<sup>7-9</sup> Here, we construct different ANN regression models by changing the number of hidden layers (1 to 4) and hidden nodes (5 to 30), to understand the effect of the number of hidden layers and hidden nodes on the R<sup>2</sup> score of the ANN model. Each layer is fully connected to its preceding layer. An example is shown in Figure S3. Each node represents a weighted linear summation function and an activation function. The activation function is ReLu in each layer. The input is the set of coordinates of CG beads, and transformed non-linearly to predict the coordinates of atoms in the all-atom model, as the output. The loss function is the mean of squared errors (MSE) between the true values and predicted ones along with the L2 penalty (alpha = 0.1). Adam algorithm with an initial learning rate of 0.001 is employed to obtain the optimized parameters in decreasing the loss function.<sup>2</sup> The effect of the number of nodes on its  $R^2$  score is shown in Figure S4. It's found that the  $R^2$  score of ANN models with two hidden layers increases drastically as the number of hidden nodes increases from 5 to 10. Whereas it changes slightly as the number of hidden nodes is further increased to 30. The number of hidden layers on ANN models (10 hidden nodes in each layer) has little impact on the R<sup>2</sup> score as shown in Figure S5. Hence we used the ANN models with two hidden layers and 10 hidden nodes in

each layer. The training and testing of ANN models and the following k-NN, gaussian process regression, and random forests were achieved by using the Scikit-learn package.<sup>10</sup>



Figure S3. A representative architecture of the ANN model used in this study.



Figure S4: The R<sup>2</sup> scores of ANN models with different number of nodes in each hidden layer





Figure S5. The R<sup>2</sup> scores of ANN models with different number of hidden layers for (a) furan,
(b) benzene, (c) hexane, (d) naphthalene, (e) graphene, and (f) fullerene. The number of nodes in each layer is 10 and the sample size is 5000.

k-NN is a simple ML model that uses the distance between data points to solve classification or regression problems.<sup>11</sup> The most used distance metric is the Euclidean distance to measure the similarity between two points. Based on this, the data points with the *k* shortest distance from the data to be predicted would be selected to assign classes or values of the unknown data. Here, we studied the performance of k-NN models with different k values (3, 5, 8). The R<sup>2</sup> score of k-NN models for testing is increased slightly as the k value is increased from 3 to 8 as shown in **Table S1**. As the k value increases, the R<sup>2</sup> score for testing is increased slightly from ~ 0.98 to ~ 0.99 for furan, benzene, and naphthalene. For hexane, it's increased much more from ~ 0.73 to ~ 0.76, by 0.03. The R<sup>2</sup> score for testing is always higher than 0.99 for graphene, while it's

around 0.97 for fullerene regardless of k values. Overall, the k value is set to be 5 in the study unless specified.

	k=	=3	k=	=5	k=8		
	training	testing	training	testing	training	testing	
Furan	0.991±0.001	0.986±0.000	0.990±0.001	0.988±0.000	0.992±0.002	0.990±0.000	
Benzene	0.989±0.002	0.987±0.000	0.987±0.001	0.989±0.000	0.991±0.001	0.991±0.000	
Hexane	0.734±0.003	0.728±0.002	0.761±0.002	0.750±0.002	0.771±0.002	0.761±0.001	
Naphthalene	0.990±0.001	0.988±0.000	0.992±0.002	0.990±0.000	0.993±0.001	0.991±0.000	
Graphene	0.992±0.001	0.991±0.000	0.990±0.001	0.993±0.000	0.992±0.001	0.994±0.000	
Fullerene	0.980±0.004	0.973±0.003	0.971±0.002	0.975±0.003	0.970±0.002	0.975±0.002	

**Table S1**. The  $R^2$  scores of k-NN models with different k values. The dataset size is 5000.

RF is an ensemble model consisting of several decision trees to predict the final labels/values by selecting a subset of features for each decision tree.<sup>2</sup> The performance of RF models is usually better than that of one single decision-tree model. The minimum number of samples splitting a node is set to be 2, and the minimum number of samples in a leaf node is 1. The number of features to consider for best splitting is the total number of features. The effects of max\_depth and n\_estimators are explored in **Figure S6.** Specifically, we studied the RF models, MSE is used to split the trees. It can be found that the RF models are more sensitive to the changes of max depths. To be specific, as the max depth increases from 5 to 10, the R<sup>2</sup> score for testing increases from around 0.72 to 0.95 for furan, naphthalene, and fullerene, with a fixed number of estimators (5 or 10 or 15). On the other hand, the R<sup>2</sup> score for testing is only increased slightly by less than 0.1 as the number of estimators increases from 5 to 20. Based on these results, we recommend using a max depth of 10 and the number of estimators of 10 for RF models.



**Figure S6.** The heatmap for R<sup>2</sup> scores of RF models for testing with various max depths and number of estimators. The panels represent RF models for backmapping CG molecules of (**a**) furan, (**b**) benzene, (**c**) hexane, (**d**) naphthalene, (**e**) graphene, (**f**) fullerene.

GPR is a nonparametric Bayesian approach to infer the probability distribution over all possible values.<sup>12</sup> In the Gaussian prior, the collection of training and testing data points are joint multivariate Gaussian distributed, which can be described in **Equation S1**.

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix} \right)$$
Equation S1

Where y is the output values of training data X,  $f_*$  is the predicted label/output of testing data X\*. K is the kernel or covariance. The constant kernel with the radial basis function (RBF) is one of the popular kernels, as shown in **Equation S2**.

$$k(x, x') = \sigma_f^2 exp(-\frac{1}{2\ell^2} ||x - x'||^2)$$
 Equation S2

Where  $\sigma$  is the signal variance, and *l* is the length scale. These two parameters are optimized during training to maximize the marginalized log-likelihood of the training data in **Equation S3**.

$$log(p(y|X)) = log N(0, K(X,X)) = -\frac{1}{2} yK^{-1}(X,X)y - \frac{1}{2}log|K(X,X) + \sigma_n^2 I| - \frac{N}{2}log(2\pi)$$
Equation S3

According to multivariate Gaussian theorem, the predicted values  $f^*$  are in a normal distribution with mean value  $\bar{f}^*$  and covariance  $\Sigma^*$ , which are shown below:

Equation S4	$f^* V \vee V^* \sim \mathcal{M}(\bar{f}^* \nabla^*)$
Equation S5	$\int [\Lambda, y, \Lambda \sim \mathcal{N}(0^{-1}, \mathbb{Z})]$
	$f^* = \mu^* + K(X^*, X)[K(X, X) + \sigma_n^2 I]^{-1}(y - \mu)$
Equation S6	$\Sigma^* = K(X^*, X^*) - K(X^*, X) [K(X, X) + \Sigma^2 I]^{-1} K(X, X^*)$

Kernel ridge regression (KRR) is one of the kernel-based regression methods. It converts the input data from a low dimensional space into a new high dimensional space by using kernel-trick.<sup>1</sup> In this study, the kernel is a three-degree polynomial kernel. SVR is the regression model as an extension of the support vector machine (SVM).<sup>2,10</sup> In this study, it's a multi-output SVR model where N ( the dimensionality of the output data) regression models were trained on N columns of the output space, respectively. Note, KRR and SVM were used to ensure that the predictions from other models are not due to overfitting. Overfitting can be referred as when a model performs quite well on the training data but fails to predict "unseen" data.<sup>2,5</sup> In general, the reason for overfitting is that a model learns all the information containing noise and fluctuations in the training data to the extent that it impairs the performance of the model on new data.<sup>2</sup>

	Benze	ene	Graphene			
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy		
KRR <sup>a</sup>	0.998	0.998	0.99	0.99		
SVR <sup>b</sup>	0.997	0.997	0.98	0.98		

**Table S2**. The testing and training accuracies of KRR and SVM models for backmapping benzene and graphene. The dataset used consists of 5000 samples.

a:Parameters for the KRR model:kernel = poly, alpha = 0.1, coef0 = , degree = 3. b:Parameters for the SVR model:sklearn.multioutput.MultiRegressor(estimator = SVR(*kernel* = '*rbf*', *degree*=3, *gamma*='*scale*', *coef0*=0.0, *tol*=0.001, *C*=1.0, *epsilon*=0.1))

#### S3. Performance of machine-learning models

The performance of ML regression models is determined by the root of mean square error (RMSE) and  $R^2$  score, which are described below.<sup>2</sup>

**RMSE**: The RMSE of predicted values/vectors  $y_{pred}$  compared with true values/vectors  $y_{true}$  is calculated by using the following equation. N represents the dimensions in the output space.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_{i,true} - y_{i,pred})^2}$$
 Equation S7

**R**<sup>2</sup> score: The R<sup>2</sup> score defined for the regression models as shown in **Equation S8**. n is the number of samples in the train/test set, and N is the dimension of output.  $\bar{y}_{i,j,true}$  is the average value of the *j*th element in the true vector **y**<sub>i,true</sub>(the true output value of the *i*th sample).

$$R^{2} = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{N} (y_{i,j,true} - y_{i,j,pred})^{2}}{\sum_{i=1}^{n} \sum_{j=1}^{N} (y_{i,j,true} - \overline{y}_{i,j,true})^{2}}$$
Equation S8



Figure S7. The CG hexane model and its two backmapped all-atom models.

**S4. Uncertainty Quantification:** To test the robustness of ML models, bootstrapping is employed to obtain the uncertainty quantifications of these ML models.<sup>1,13,14</sup> Bootstrapping is a widely used method to resample the dataset.<sup>13,15</sup> Here, we resampled the training dataset with replacement for 500 times, and each resampled dataset was used to build ML models. As a result, 500 ML models could be built and were tested against the testing dataset to calculate R<sup>2</sup> score. The histograms of these R<sup>2</sup> scores were plotted with 95% confidence interval and their average values shown in the histogram.



**Figure S8**. Uncertainty quantification of the testing R<sup>2</sup> scores of the models: (1) ANN, (2) k-NN, (3) GPR, and (4) RF models for (a) benzene, (b) hexane, (c) naphthalene, (d) graphene, and (e) fullerene.



Figure S9. Uncertainty quantification of the training  $R^2$  scores of the models: (1) ANN, (2) k-

NN, (3) GPR, and (4) RF models for (a) furan, (b) benzene, (c) hexane, (d) naphthalene, (e) graphene, and (f) fullerene.

## **References:**

- 1 S. K. Singh, K. K. Bejagam, Y. An and S. A. Deshmukh, *J. Phys. Chem. A*, 2019, **123**, 5190–5198.
- 2 P.-N. Tan, M. Steinbach, A. Karpatne and V. Kumar, *Introduction to Data Mining*, Pearson Education, 2019.
- 3 W. Jiang, D. J. Hardy, J. C. Phillips, A. D. Mackerell Jr, K. Schulten and B. Roux, *J. Phys. Chem. Lett.*, 2011, **2**, 87–92.
- 4 J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kalé and K. Schulten, *J. Comput. Chem.*, 2005, **26**, 1781–1802.
- 5 T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning; Data mining, Inference and Prediction*, Springer Verlag, New York, 2001.
- 6 O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed and H. Arshad, *Heliyon*, 2018, **4**, e00938.
- J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. de Fabritiis, F. Noé and C. Clementi, *ACS Cent. Sci.*, 2019, **5**, 755–767.
- 8 W. Wang and R. Gómez-Bombarelli, *npj Computational Materials*, **5**, 1–9.
- 9 H. Chan, M. Cherukara, T. D. Loeffler, B. Narayanan and Subramanian K R, *npj Computational Materials*, 2020, **6**, 1–9.
- 10 F. Pedregosa, G. Varoquaux, A. Gramfort, et al., *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
- 11 Z. Zhang, Ann Transl Med, 2016, 4, 218.
- 12 C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2006.
- 13 Z. Li, N. Omidvar, W. S. Chin, E. Robb, A. Morris, L. Achenie and H. Xin, *J. Phys. Chem. A*, 2018, **122**, 4571–4578.
- 14 Y. An, S. Singh, K. K. Bejagam and S. A. Deshmukh, *Macromolecules*, 2019, **52**, 4875–4887.
- 15 T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, Springer Science & Business Media, 2009.