JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

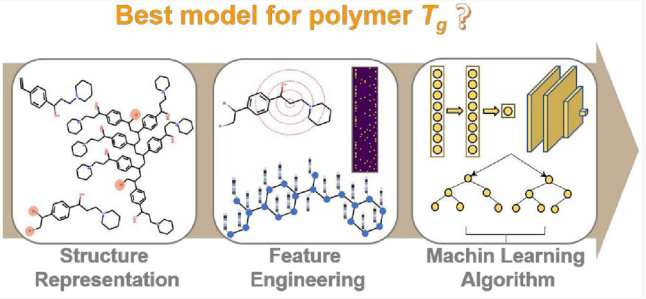
Benchmarking Machine Learning Models for Polymer Informatics: An Example of Glass Transition Temperature

Lei Tao, Vikas Varshney, and Ying Li*

Cite This: J. Chem. Inf. Model. 2021, 61, 5395−5413 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: In the field of polymer informatics, utilizing machine learning (ML) techniques to evaluate the glass transition temperature T_g and other properties of polymers has attracted extensive attention. This data-centric approach is much more efficient and practical than the laborious experimental measurements when encountered a daunting number of polymer structures. Various ML models are demonstrated to perform well for T_g prediction. Nevertheless, they are trained on different data sets, using different structure representations, and based on different feature engineering methods. Thus, the critical question arises on selecting a proper ML model to better handle the T_g prediction with generalization ability. To provide a fair comparison of different ML techniques and examine the key factors that affect the model performance, we carry out a systematic benchmark study by compiling 79 different ML models and training them on a large and diverse data set. The three major components in setting up an ML model are structure representations, feature representations, and ML algorithms. In terms of polymer structure representation, we consider the polymer monomer, repeat unit, and oligomer with longer chain structure. Based on that feature, representation is calculated, including Morgan fingerprinting with or without substructure frequency, RDKit descriptors, molecular embedding, molecular graph, etc. Afterward, the obtained feature input is trained using different ML algorithms, such as deep neural networks, convolutional neural networks, random forest, support vector machine, LASSO regression, and Gaussian process regression. We evaluate the performance of these ML models using a holdout test set and an extra unlabeled data set from high-throughput molecular dynamics simulation. The ML model's generalization ability on an unlabeled data set is especially focused, and the model's sensitivity to topology and the molecular weight of polymers is also taken into consideration. This benchmark study provides not only a guideline for the T_g prediction task but also a useful reference for other polymer informatics tasks.



1. INTRODUCTION

Polymer informatics that utilizes a data-driven approach to evaluate the physical properties of polymers is an emerging field in polymer science and engineering.[1−3] A significant challenge in polymer chemistry is investigating the almost infinite chemical space of polymers, as polymer molecular structures stem from wide varieties of chemical structures and polymerization processes.[4−6] In this context, traditional experimental-driven approaches become impractical to search or optimize the infinite polymer structures toward a set of desired objective properties. As an exciting alternative, utilizing machine learning (ML) techniques and the increasing amount of polymer data sets offer a new opportunity to tackle the challenge in the polymer field. Successful polymer informatics attempts have touched upon the a number of property predictions like polymers' electronic bandgap,[7,8] dielectric constant,[9] refractive index,[10] etc., but a lot more attention has been paid to the prediction of polymers' glass transition temperatures.[8,11−23] This is primarily reflective of the facts that (1) the glass transition temperature is an important property controlling the phase transition and therefore the application of polymers[24] and (2) the glass transition temperature T_g is the most reported experimental measurement in publicly accessible databases like PoLyInfo,[25] the Polymer Property Predictor and Database (NIST),[26] and the CROW Polymer Properties Database.[27]

Currently, one of the largest databases, PolyInfo, contains more than 13 000 homopolymers that are reported with detailed structure and property information. This homopolymer data set is the subject of our study. Among the 13 000 homopolymers, 6923 of them have been reported for measured T_g (referred to as the labeled data, Data set_1 in the manuscript) through either differential scanning calorimetry or thermomechanical analysis, while for other 5690 polymers, no

Received: August 25, 2021
Published: October 18, 2021

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

measured T_g have been reported (referred to as the unlabeled data, Data set_2 in the manuscript). If we were to obtain the T_g for all unlabeled polymer structures one by one through experiments, it would be an intimidating and unsurmountable task to complete considering the labor, cost, and time required. Here, ML algorithms are especially suitable for this task. An ML model can discover the structure−property correlation within the labeled data (Data set_1) and then apply it to the unlabeled data (Data set_2) to make predictions for their T_g. Such a data-centric approach is a much more efficient approach for the T_g estimation, in comparison to laborious experiments or molecular simulations.[28]

Establishing an ML model for polymer informatics usually involves three main steps: determining a suitable structure representation for polymers, utilizing a appropriate feature representation, and implementing a proper ML algorithm to fit the data.

Step 1. *Determining a proper structure representation.* As polymers are long-chain molecules that are difficult to be represented completely, they are usually represented by their monomeric counterpart or their repeat unit. A monomer is a small organic molecule from which a polymer is synthesized, while a repeat unit is the repeating part of the long polymer chain. The main difference between the two is that the repeat unit contains information about the bond connectivity (polymerization point) along the polymer chain. Oligomers corresponding to several repeat units chained together even possess more substructures that cannot be found in a single repeat unit. Different structure representations are therefore denoted differently under the simplified molecular input line entry system (SMILES) notation.[29−31] For example, the structure representation for poly(non-1-ene) could be its monomer "CCCCCCCC=C", its repeat unit "*C(C*)-CCCCCCC", or its oligomers of three repeat units chained together "CCCCCCCC(*)CC(CCCCCCC)CC(C*)-CCCCCCC". Note that the symbol "*" denotes the polymerization or connecting point for the repeat unit.

Step 2. *Utilizing a proper feature representation.* When the structure representation is determined, feature engineering aims to extract structural features that are most relevant to polymer properties. The most commonly used feature representation is the Morgan fingerprint that detects all substructures in the molecule.[32] Descriptors—a different feature representation—calculate constitutional, topological, or geometrical indices for polymer structures.[33] In addition, molecular embedding[34] and molecular graph[35] are also able to extract features in different ways.

Step 3. *Implementing a proper ML algorithm to process the data.* ML algorithms range from linear regression methods like least absolute shrinkage and selection operator (LASSO) to nonlinear methods such as support vector machine (SVM), feed-forward neural networks (FFNN), convolutional neural networks (CNN), Gaussian process regression (GPR), random forests (RF), etc. Various ML algorithms offer many options for model building, and usually, a reasonable choice largely relies on the researchers' domain knowledge, experience, and caution as well as available data.[36]

Extensive studies have been done for the T_g prediction of polymers, and researchers have made different choices for Steps 1−3. For example, Miccio et al.[12,13,23] (1) focused on the monomer structure of polymers (2) converted the monomer's SMILES into binary matrices with the one-hot encoding algorithm;[37] and (3) used the FFNN or CNN method to build ML models. Their ML models showed a relative error of about 6% and observed a reasonable generalization ability on about 300 polymers from different classes. Not using monomers or neural networks, Ramprasad et al.[8,38−40] (1) represented polymers by their repeat units; (2) calculated three hierarchical levels of descriptors, including atomic level descriptors, quantitative structure−property relationship (QSPR) descriptors, and morphological descriptors; and (3) utilized the GPR model in their Polymer Genome platform.[8] Their model setup led to a prediction performance of R^2 = 0.92 for T_g prediction. Very differently, Díaz et al.[16] (1) modeled each polymer with a trimeric structure that is composed of three repeating units chained together; (2) calculated 12 descriptors including surface area, volume, partition coefficient log *P*, refractivity, polarizability, and mass, etc.; and (3) used a three layers FFNN model. They achieved a good model performance of R^2 = 0.964 on their data set of 88 high-molecular-weight polymers.

In addition to the ML models mentioned above, other choices have also been investigated by researchers such as SVM,[20,41] recurrent neural networks (RNN),[16,42] and RF,[21] etc. Many models claim to be able to produce good T_g predictions, and some of them have been compared together. For example, Luo and co-workers[43] (1) represented polymers using a two-monomer structure (through which the bonding information between neighboring monomers was also included); (2) used different types of polymer representations like Morgan fingerprint, molecular embedding, and molecular graph; and (3) compared RF, SVM, and FFNN models in terms of T_g prediction. Their best ML model was found to be the SVM model using molecular embedding, with prediction performance with R^2 = 0.865. In our recent study,[44] we (1) used repeat unit containing bonding position to represent polymers; (2) improved the Morgan fingerprint representation to consider the number of each substructure in polymer molecules, instead of only marking the presence or absence of each substructure; and (3) compared LASSO, FFNN, and CNN models on Morgan fingerprint, descriptor, 2D images, and our improved Morgan fingerprint. Our FFNN model using the improved Morgan fingerprint demonstrated a good prediction and generalization ability. We obtained an R^2 of 0.85/0.83 on training/test sets and further validated the model's generalization ability with extra data points from experiments and molecular dynamics simulations.[44]

While various ML models have been formulated for polymer's T_g prediction, there are still questions that remain to be answered regarding the choices in each step. (1) What polymer representation is the most appropriate to use? Researchers have represented polymers by monomer structure, two-monomer structure, repeat unit, or trimeric structure composed of three repeat units, etc. Their effect on the model performance is not clear and is not easy to answer. (2) What feature representation is the most appropriate one to utilize? Carefully selected descriptors, circular Morgan fingerprint, molecular embedding, molecular graph, etc. are all able to serve as the input feature to the ML model, but there is no direct comparison for all of them regarding their influence on T_g prediction. (3) What ML algorithm is the most proper one to implement? FFNN, CNN, RF, SVM, GPR, etc. have been found to perform well for the T_g prediction. A fair model comparison would be highly preferred so that we know which algorithm can lead to the best T_g prediction with generalization ability for unlabeled data set.

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

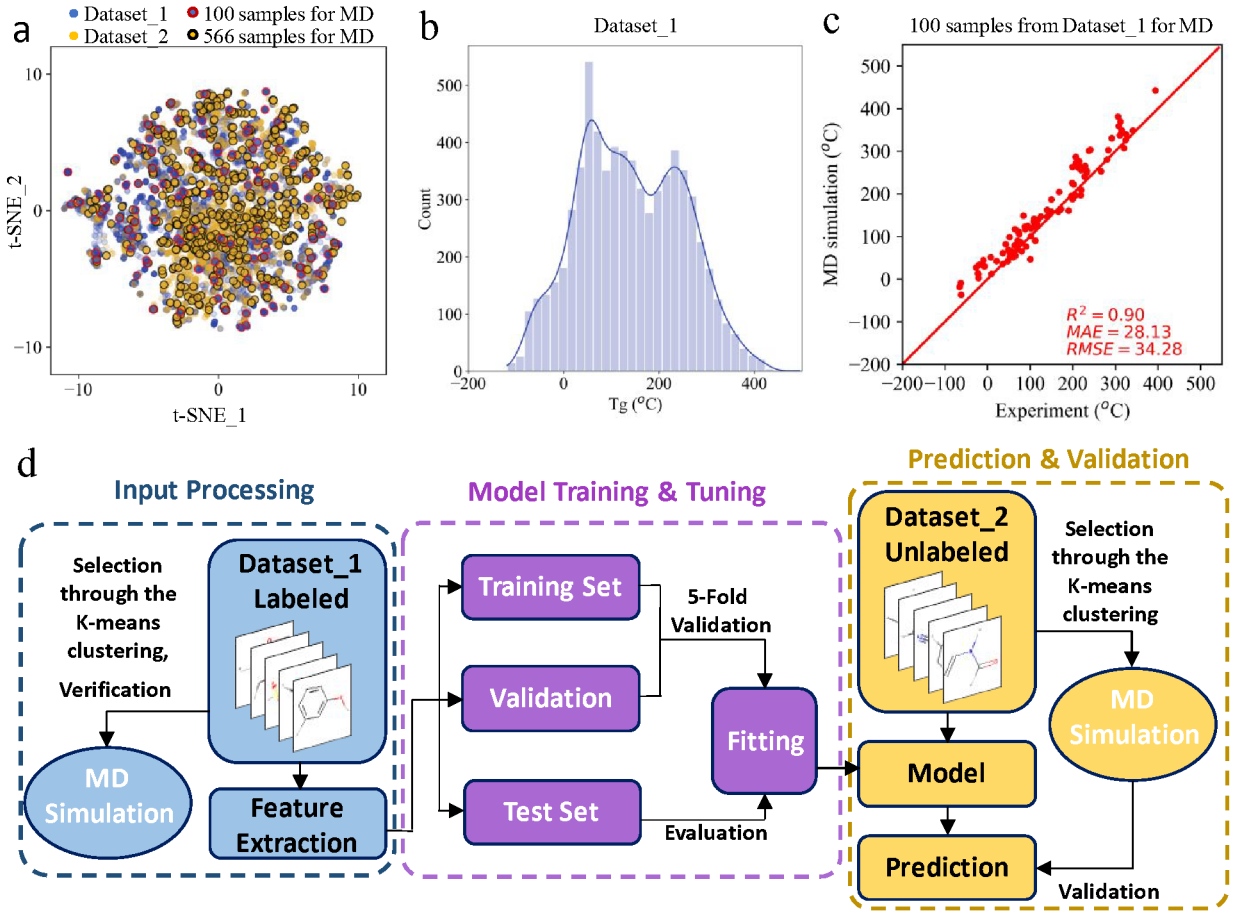


Figure 1. Workflow for machine learning of polymer's T_g prediction. (a) Chemical space visualization of Data set_1 (blue), 100 simulated polymers from Data set_1 test set (blue with the red edge), Data set_2 (orange), and 566 MD simulated polymers from Data set_2 (orange with the red edge) using the t-SNE algorithm. (b) T_g distribution of the labeled Data set_1, collected from the PolyInfo. Its T_g ranges from −118 to 495 °C. (c) Parity plot of MD simulated T_g vs experimental values for the 100 simulated polymers from Data set_1 test set. The comparison between experimental values and MD simulations gives R^2 = 0.90, MAE = 28.13, and RMSE = 34.28. (d) The labeled Data set_1 is used for ML model training and evaluation, and the unlabeled Data set_2 is used as an extra data set for the further assessment of the ML model's generalization ability. ML model training is carried out with 5-fold cross-validation and a holdout test set is used to evaluate the trained model. MD simulations are verified using the labeled Data set_1 and can validate the ML performance using the unlabled Data set_2.

In order to answer these important questions, this work carries out a systematic benchmark study to evaluate the ML model performance on T_g prediction. We assess different polymer representations, feature representations, and ML algorithms together following a supervised learning process, as shown in Figure 1d. First, the labeled Data set_1 is split into three groups: training, validation, and test sets. An ML model is fitted on the training set to find the pattern and establish the structure−property correlation. Second, as the fitting randomness and hyperparameters tuning affect the prediction performance, the validation set serves to monitor the prediction accuracy when optimizing a specific ML model. Third, the ML model is further evaluated using the test set, which has not been seen by the model during training and tuning. Thus, a trustworthy model performance is obtained. Furthermore, we evaluate the generalization ability of the obtained ML model based on the high-throughput molecular dynamics (MD) simulations of 566 selected polymers from the unlabeled Data set_2 through the K-means clustering. After comparing 79 model setups using 3 different polymer representations, 7 feature representations, and 8 ML algorithms, we reveal the pros and cons behind different options. Such a systematic benchmark study provides valuable guidance for the model selection in polymer's T_g prediction, and it would also benefit other polymer informatics tasks such as mechanical, electronic, or optical properties.[45−48]

2. DATASETS, MODELS, AND METHODS

2.1. Data Sets. We have collected a large data set of polymer molecules from the PoLyInfo database,[25] as detailed in our recent study.[44] Both the T_g values and molecular structures are known for 6,923 homopolymers, which make up the labeled Data set_1. Its T_g ranges from −118 to +495 °C, the distribution of which is presented in Figure 1b. Another 5675 homopolymers compose the unlabeled Data set_2, whose T_g is not available or reported. The t-SNE plot in Figure 1a compares the chemical space occupied by Data set_1 and Data set_2.[44] It suggests that these two data sets share similar chemical space. We adopted the *K*-means clustering (K = 600) for data partition in Data set_2 for subsequently selecting 600 polymers for MD simulations to ensure that polymers for which MD simulations are performed are well-scattered and representative of the whole chemical space as well, as shown in Figure 1a. Because of the complexity of some structures, the

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

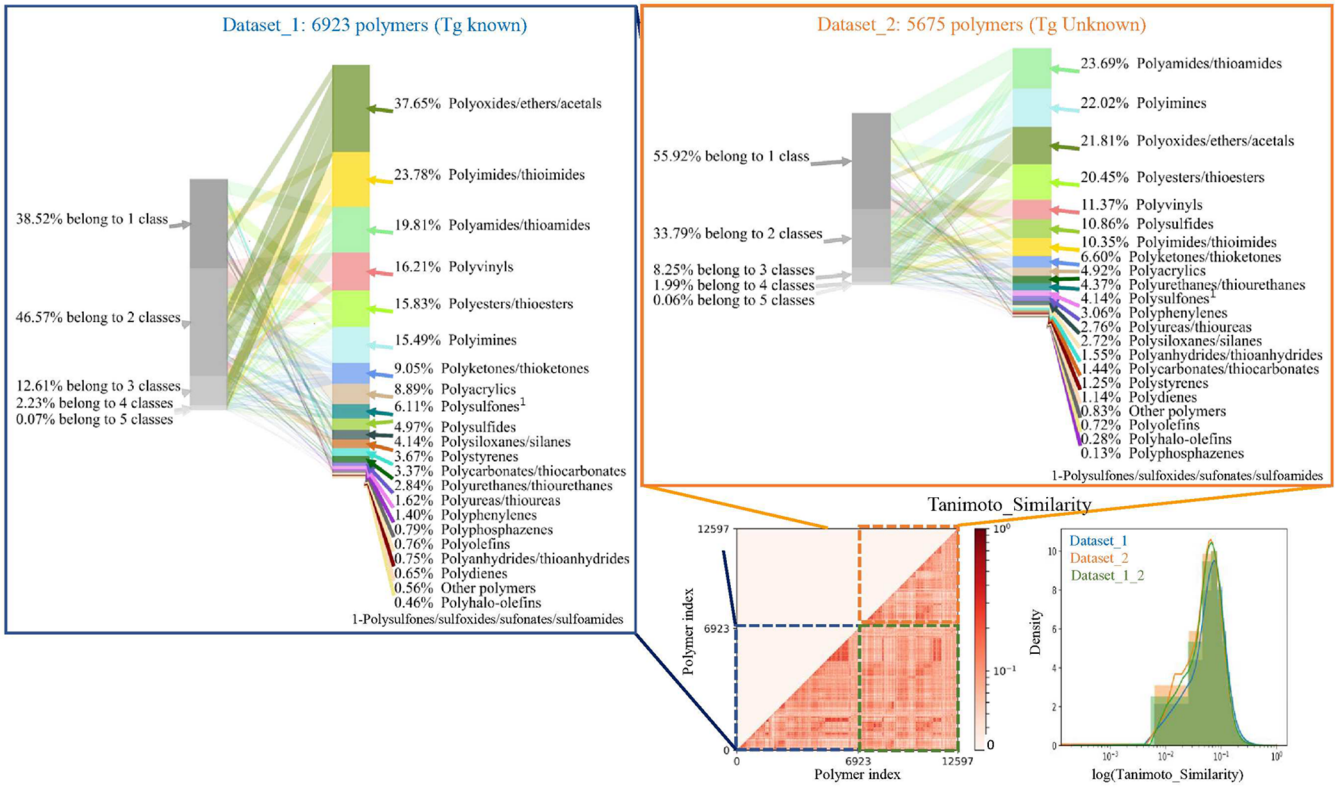


Figure 2. Comparison of the Data set_1 and Data set_2. Both data sets contain common polymer classes. The dominant polymer classes in Data set_1 also constitute the majorities in Data set_2. The pairwise Tanimoto similarity (Tc) between polymers is displayed in a matrix and a histogram plot. Two polymers are the same if Tc = 1 and totally different if Tc = 0. The Tc between any two polymers is mostly concentrated around 0.2 irrespective of how polymers are compared between 2 polymers in Data set_1, 2 polymers in Data set_2, or between 1 polymer in Data set_1 and one polymer in Data set_2.

MD simulations for 566 structures out of 600 selected polymers are successfully performed. The workflow in Figure 1d shows the roles of Data set_1 and Data set_2, respectively. Note in addition to the typical training process using Data set_1 and the prediction process using Data set_2, we use high-throughput MD simulations as further evaluation of an ML model's generalization ability (for which K-means clustering was peformed as mentioned above). To verify the accuracy of MD simulations, we also select 100 polymers from the test set of Data set_1 through the *K*-means clustering. Both of their chemical structures and experimental T_g are known, making them a suitable baseline to verify our MD simulations. Figure 1a shows that the selected polymers are well-scattered and representative of the whole chemical space. Figure 1c demonstrates that the MD simulations agree well with the experiments. Despite the uncertainties involved in MD simulations, reasonable estimations are obtained with R^2 = 0.90, MAE = 28.13, and RMSE = 34.28.

Unlike the small data sets of specific polymer classes, our Data set_1 and Data set_2 cover large classes of polymers, as demonstrated in Figure 2. The dominant polymer classes in Data set_1 also constitute the majorities in Data set_2. For example, more than one-third (37.65%) of the polymers in Data set_1 belong to polyoxides/ethers/acetals, accounting for about one-fifth (21.81%) of the polymers in Data set_2. The proportions of the main classes in each data set are roughly the same, making it possible to generalize an ML model from Data set_1 to Datset_2. Besides, the use of large and diverse data sets (22 polymer classes in total) is critical for pattern recognition. Otherwise, the trained ML model can only apply to certain classes of polymers in the local chemical space.[23,49,50]

To better understand the similarity or difference between polymers in Data set_1 and Data set_2, we calculate the structural similarity coefficient—the Tanimoto coefficient[51]—for each pair of polymers with the following steps: 1. The repeat unit of each polymer in both Data sets is transformed into Morgan fingerprint[32] using radius 3 and 2048 bits through RDKit.[52] 2. Tanimoto coefficient Tc[51] is calculated pairwise by comparing the fingerprints (substructures) of two polymers. The coefficient is defined as the ratio of the number of substructures common to two polymers to the total number of substructures present in both of them (eq 1). Two polymers are the same if Tc = 1 and totally different if Tc = 0. Figure 2 shows the pairwise similarity coefficient that is calculated (1) between two polymers in Data set_1, (2) between two polymers in Data set_2, and (3) between one polymer in Data set_1 and one polymer in Data set_2. The summarized histogram in Figure 2 indicates that the obtained Tc is around 0.2 for our collected polymers regardless of how we group them into different data sets. Most polymers are found not similar to each other, and this diversity is helpful for our ML model training and validation. ML Models trained on diverse data sets are more likely to have better generalization ability.[53] A similar Dice coefficient is defined by eq 2, and a similar result can be found in the Supporting Information, Figure S1.

$$\text{Tanimoto coefficient Tc} = \frac{c}{a + b + c} \quad (1)$$

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

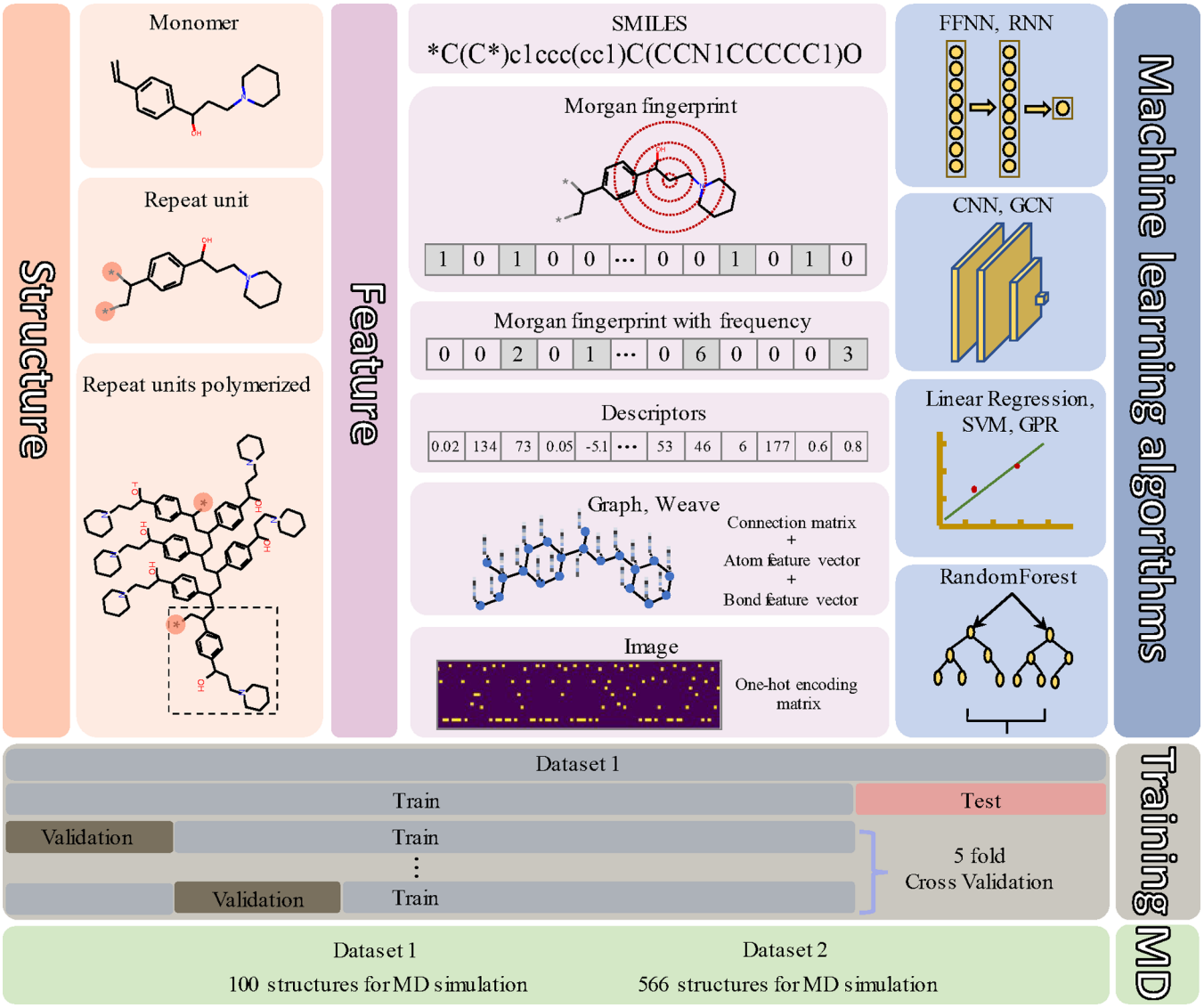


Figure 3. Three major steps for setting up a proper ML model for polymer's T_g prediction. It shows the different options we have in each step, from (1) structure representation to (2) feature representation and to (3) ML models. Going through the three steps leads us to a specific model setup. For example, the path of "Monomer → Morgan Fingerprint → FFNN" means using monomer as the polymer structure representation, then calculating the Morgan fingerprint from the monomer structure, and finally feeding the feature vector into a FFNN model for model training. A total of 79 different paths (namely 79 different model setups) are investigated in this benchmark study.

$$\text{Dice coefficient Dc} = \frac{2c}{(a + c) + (b + c)} \quad (2)$$

a is the count of bits "on" in polymer A's fingerprint but not in polymer B's fingerprint. b is the count of bits "on" in polymer B's fingerprint but not in polymer A's fingerprint. c is the count of the bits "on" in both A's and B's fingerprints.

2.2. Structure Representations. Monomer and repeat unit are two easy-to-use representations of polymers. As aforementioned, monomer, repeat units, trimer, two-monomer structures, etc. have been used in the literature as polymer structure representations. In this study, we prepared repeat unit structure, monomer structure, and structure corresponding to multiple repeat units chained together for each polymer. Figure 3 uses the structure of poly(non-1-ene) as schematic diagrams in which its monomer "CCCCCCCC=C", repeat unit "*C(C*)CCCCCCC", and oligomer of 16 repeat units chained together "CCCCCCCC(*)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(CCCCCCC)CC(C*)CCCCCCC" are listed together. It should be emphasized that the SMILES strings for monomer, repeat unit, and oligomer are obtained by carefully processing the molecular structure through RDKit.[52] The simple repeating of strings does not comply with SMILES rules for valid molecules. We should emphasize that (1) Monomer is the reacting molecule in the polymerization process. Its chemical structure is simple and clear, but the same monomer can polymerize into different polymer structures via different reactions, such as addition and condensation polymerizations. For example, monomer buta-1,3-diene can polymerize into polyethene or poly(but-1-ene) through different addition polymerization reactions.[25] (2) The repeat unit structure is unique to each polymer, and it contains a

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

special "*" symbol to indicate the polymerization point. When the repeat unit is cut out from the original polymer chain, the substructures near the connecting "*" are not as complete as in the original long-chain structure. (3) Chaining several repeat units together may be a better way to represent polymers, as it retains more substructures in the original long-chain structure, especially the ones near connecting point. Determining the proper structure representation is the first step in a polymer informatics task.

2.3. Feature Representations. Once the appropriate structure representation is determined, the next step is to extract numerical features based on the structure representation and use it as the input to an ML model. There are mainly seven different feature representations for polymer structures as illustrated in Figure 3 (see Supporting Information, Section S2 for more details of feature representations). (1) The SMILES notation[29] of the repeat unit—as a string sequence input—can be processed and fed into a RNN model. (2) The Morgan fingerprint (MF) algorithm identifies all the substructures in a molecule and marks them in a bit vector based on the existence of each substructure. This feature representation is in a format of vectors that are flexible to be fed into various ML models. (3) We improved the default Morgan fingerprint to also consider the number of each identified substructure as considering substructures' frequency of occurrence adds more information to the extracted feature vector, which will be referred to as Morgan fingerprint with frequency (MFF). (4) Molecular embedding (ME) learns vector representations with continuous values based on substructures. The sparseness of MF and MFF can be avoided in the ME vector representation obtained using the package Mol2Vec.[34] (5) Descriptor calculations are supported in RDKit for 200 descriptors (see Supporting Information, Section S2 for the list of 200 RDKit descriptors utilized), or supported in alvaDesc[54] for more than 6000 descriptors. Physical and chemical characteristics calculated from molecular structures constitute a vector of feature representation. The additional calculation for descriptors (such as quantum chemistry informed descriptors) require more effort and time than the Morgan fingerprint.[33] (6) Molecular graph (MG) manages the molecular structure as an undirected graph. The vertices and edges of the graph correspond to the atoms and chemical bonds. The MG representation is designed for graph convolutional neural networks (GCNN) exclusively. (7) 2D image representations can be obtained through the transformation of SMILES notation. Based on a predefined dictionary of SMILES characters, one-hot encoding algorithm[37] converts each polymer's SMILES into a 2D binary matrix. This representation serves as a practical fit for a 2D CNN model similar to the image recognition problem.[12,44] Besides, BigSMILES, a line notation that supports the intrinsically stochastic nature of polymers on top of the SMILES, has been proposed for the representation of polymers as well.[55] Nevertheless, BigSMILES is not yet supported by the Cheminformatics packages, such as RDKit. Therefore, we did not consider BigSMILES for the representation of polymers in this benchmark study.

2.4. ML Models. Different feature representations are suitable for one or several ML algorithms and it is up to researchers to use their discretion in selecting the proper ML algorithm. In the following, we give a brief overview of the popular ML algorithms adopted in this study. We have implemented all of them, and the parameter settings can be found in the Supporting Information, Section S3.

2.4.1. Feed-Forward Neural Networks (FFNN). FFNN is composed of a set of neurons connected layer by layer. Each neuron works as a function to accept inputs and generate an output, sometimes followed by a nonlinear activation function like rectified linear unit (ReLU). Our FFNN architecture has two hidden layers between the input and output layers. Each layer contains eight neurons, and the ReLU activation function is used, as discussed in our recent study.[44] We implement the FFNN model using the Keras package.[56]

2.4.2. Recurrent Neural Networks (RNN). RNN contains neurons that accept sequential data such as characters or words, whose original purpose in natural language processing is to predict the next tokens in the sequence given past tokens. In our case, we treat the SMILES notation of polymers as the input sequence to the RNN model and then specify the T_g as the output token to be predicted. Our RNN model is realized and introduced in our recent study.[57]

2.4.3. Convolutional Neural Networks (CNN). CNN architecture contains fully connected layers as in FFNN and RNN, and also has convolutional layers.[58] A filter matrix slides over the input in convolutional layers and extracts features at different levels. The shape of a filter matrix can be arbitrary so that it can be applied to any input format such as 1D vectors, a 2D matrix, or even a 3D matrix. Our feature representations like fingerprint, descriptors, and molecular embedding are in the form of 1D vectors, making 1D CNN a good model option to utilize. In our previous study, we have examined CNN on polymer's 2D images obtained through one-hot encoding, but the 2D CNN model was found not to be performing well.[44] Our CNN model is implemented using the Keras package.[59]

2.4.4. Graph Convolutional Neural Networks (GCNN). GCNN is a generalization of the CNN model on undirected graphs.[60] It takes in a graph whose vertices and edges correspond to atoms and chemical bonds. A filter is then sliding over localized atoms and chemical bonds to aggregate their attribution vectors together. When different filters slide on different regions, the features at various levels are aggregated and extracted. At last, the GCNN converts a molecular graph into a feature vector that possesses complete information about the molecule. We employ DeepChem package[61] to build the GCNN model on our data set.

2.4.5. LASSO Regression. LASSO is a linear regression model that adds an L1-norm regularization on the weights.[62−64] It inclines to produce zero regression coefficients and eliminate unimportant features from the model. We use the LASSO algorithm to examine the linear structure−property relationship, and the algorithm is implemented with the Scikit-learn package.[65]

2.4.6. Support Vector Machine (SVM). SVM is different from linear regression that minimizes the sum of squared error. It has the flexibility to tolerant some errors and finds a proper fitting function. The use of the kernel function supports efficient data mapping to a higher dimension. Commonly used kernel functions include linear kernel, polynomial kernel, sigmoid kernel, etc. Our SVM uses a linear kernel in the Scikit-learn package.[65]

2.4.7. Gaussian Process Regression (GPR). GPR is a nonparametric approach that calculates the distribution over all possible functions that fit the observed data. It measures the similarity between points based on kernel functions to make predictions for new data. Kernel functions can be Matern

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

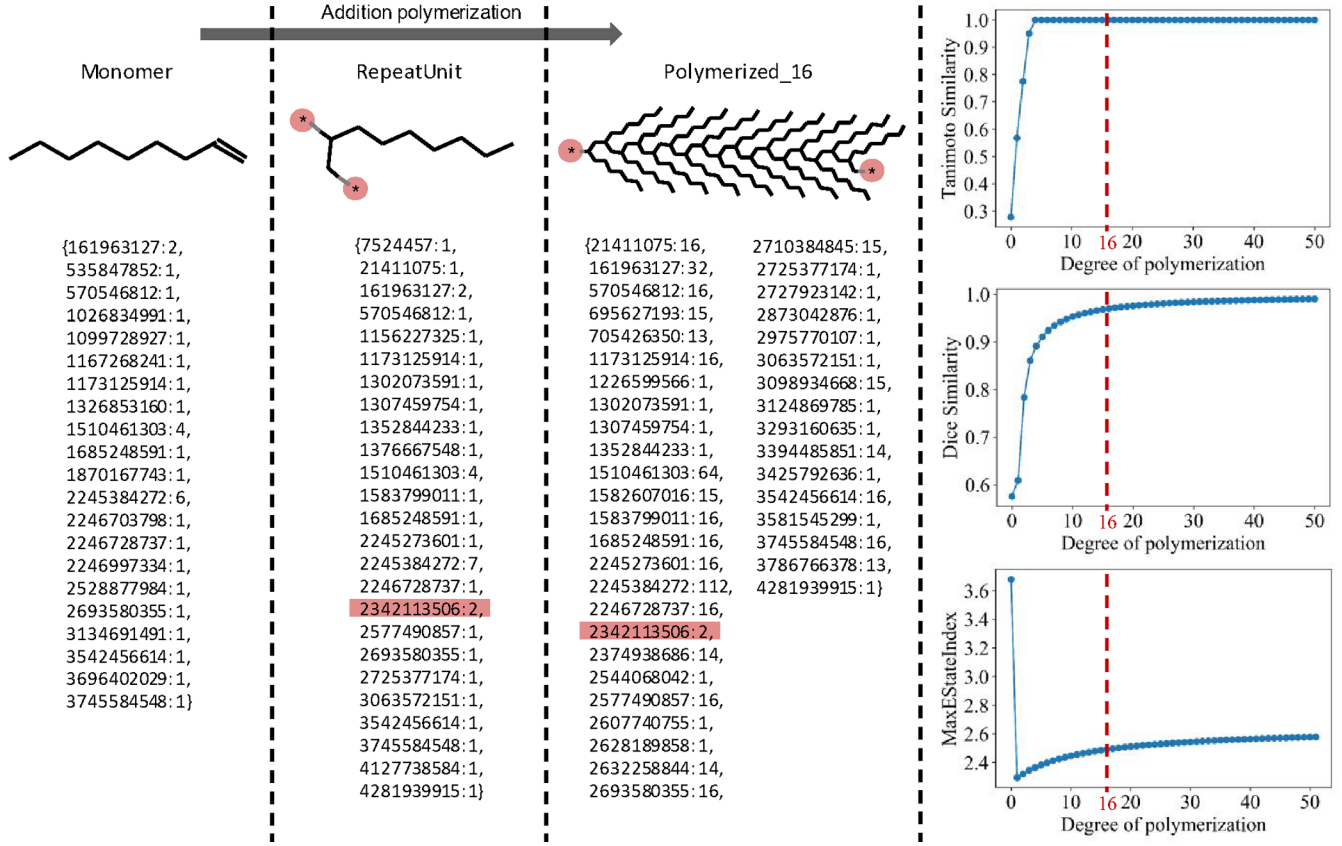


Figure 4. Convergence of polymerization degree in terms of substructures and descriptors. There are 51 structures examined in the list [Monomer, RepeatUnit, Polymerized_2, ..., Polymerized_50]. The substructure identifiers are given under each structure representation with their frequencies of occurrence. The Tanimoto similarity and Dice similarity between neighboring structures are plotted. The value of the descriptor MaxEStateIndex for each structure representation in the list is plotted as an example. A convergence pattern is noticed in each plot, and Polymerized_16 is considered as a long enough oligomer in terms of substructures and descriptors.

kernel, white kernel, radial bases function (RBF) kernel, etc. We use a combination of white kernel and RBF kernel within the Scikit-learn package.[65]

2.4.8. Random Forests (RF). RF uses the ensemble learning method for regression. It combines results from multiple decision tree regression. While each tree generates its prediction, the prediction based on averaging n-different trees has better accuracy than any single tree. We tune the number of individual trees to be n = 100 in our RF model, considering the balance of predictive accuracy and computational cost. The RF algorithm is implemented using the Scikit-learn package.[65] Other ensemble methods are also applicable such as AdaBoost, LSBoost, or Gradient Boosted decision trees. This study takes RF as the representative ensemble model and did not test all methods, as we expect that these ensemble methods will have similar performance.

2.5. Model Training and Evaluation. We have introduced the three major steps for setting up a proper ML model for a polymer's T_g prediction. Figure 3 displays the options we have in each step from (1) structure representation to (2) feature representation and to (3) ML algorithms. Going through the three steps leads us to a specific ML model setup. For example, the path of "Monomer → Morgan Fingerprint → FFNN" means using monomer as the polymer's structure representation, then calculating Morgan fingerprint from the monomer structure, and finally fitting the feature vector with FFNN algorithm for model training. In total, we investigate 79 different paths (namely 79 different model setups) in this benchmark study. The training process for each model setup follows the same predefined strategy using Data set_1. We first split the data set into an 80% training set and a 20% test set. Then, we further divide the training set into five subsets and carry out 5-fold cross-validation to obtain a reliable evaluation of the trained model. Finally, the test set is used to evaluate the performance of the model on previously unseen data (see Supporting Information, Section S4 for the examination of model performance vs training set size). We compute three measures to evaluate the predictive performance: determination coefficient R^2, mean absolute error (MAE), and root-mean-square error (RMSE). Generally a high R^2 in our results corresponds to lower MAE and RMSE. To make direct comparison with others' model in which R^2 is usually reported, we mainly use R^2 in our following comparison.

2.6. Model Validation with MD Simulations. Because Data set_2 contains polymers new to Data set_1, the performance of the obtained ML model on Data set_2 is of key interest. Toward that, we use predicted T_g values from 566 MD simulations for a subset of Data set_2 polymers as the baseline to demonstrate the ML model's generalization ability. For each selected polymer, we first build a single chain having ~2000 atoms for energy minimization. Then we construct an amorphous cell consists of 20 polymer chains through the self-avoiding random walks in space.[66] The homogeneously packed cubic cell has a dimension around 150 Å. Periodic boundary conditions are applied, and the system is equilibrated first with a 21-step molecular dynamics equilibration protocol[67] using

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

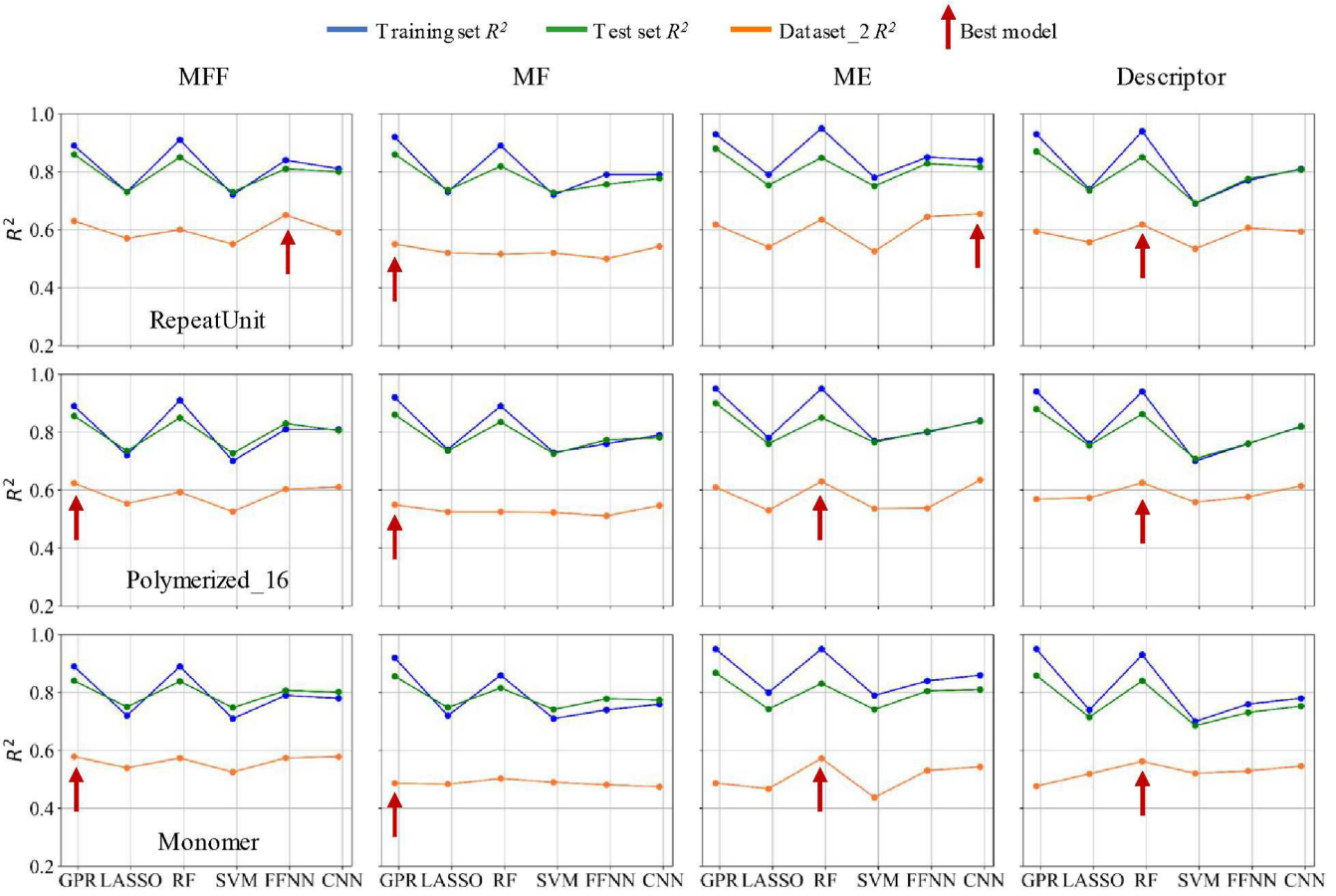


Figure 5. Performance of ML models. Models in the same row are using same structure representation, like RepeatUnit (top), Polymerized_16 (middle), and monomer (bottom). Models in the same column are based on same feature engineering, like MFF, MF, ME, and Descriptor. R^2 for training set, test set, and Data set_2 are plotted in blue, green, and orange, respectively. The best models are selected based on R^2 on Data set_2 and are indicated by the red arrows.

the PCFF force field.[68−71] PCFF (polymer consistent force-field) is a second-generation force field,[72−76] which has been parametrized against a wide range of experimental observables for organic compounds containing H, C, N, O, S, P, halogen atoms, and ions. PCFF has a broad coverage of organic polymers in calculations of cohesive energies, mechanical properties, compressibilities, heat capacities, elastic constants. Thus, the PCFF force field is particularly suitable for the molecular simulation of polymer's T_g value.[71] A cooling process simulation generates the specific volume vs temperature curve, from which rubbery phase and glassy phase are identified, and their intersection gives the T_g value.[77−79] It is acknowledged that MD conditions are not exactly consistent with experiments such as the MD's high cooling rate on the nanosecond time scale.[78,80−82] Nevertheless, a consistent trend between MD simulated T_g and experimental observation has been demonstrated by Afzal et al.[83] over 315 common polymers. To verify the reliability of our MD simulations, we also selected 100 polymers in our Data set_1 test set for MD simulations through K-means clustering (K = 100). We obtained R^2 = 0.90, MAE = 28.13, and RMSE = 34.28, demonstrating that our MD simulated T_g agrees well with the experimental values in the range of uncertainties, as given in Figure 1c. With reliable MD simulation results, when evaluating the performance of an ML model, we treat the 566 MD simulations on Data set_2 as the most important reference to compare against. The model performance on Data set_2 is more critical than on the test set and training set, because a different data set can better validate the generalization ability of an ML model.

3. RESULTS AND DISCUSSION

3.1. Convergence of the Degree of Polymerization in Structure Representation. We first examine the convergence of the degree of polymerization using a repeat unit. Monomers, repeat units, and oligomers corresponding to several repeat units chained together are representatives of long-chain polymers.[11,16,43] Compared to monomers, the repeat unit contains bonding information indicated by the "*" symbol in their SMILES representation, which explicitly suggests the polymerization point of a polymer. Chaining several repeat units together incorporates more structure information in the originally long-chain polymer, but whether an oligomer is better than a monomer or a repeat unit remains ambiguous. Therefore, we built a list of 51 structures for the same polymer to examine their convergence.

Taking the poly(non-1-ene) as an example, a monomer, a repeat unit, and an oligomer are represented in Figure 4. We indicate the monomer as zero degree of polymerization on the x axis. This small molecule is far away from being a long-chain polymer. The single repeat unit is indicated as one degree of polymerization. Similarly, the "Polymerized_16" indicates an

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

oligomer that corresponds to 16 repeat units chained together. Comparing the structures of different polymerization degrees give us a clear indication of convergence pattern in Figure 4. In a list of [Monomer, RepeatUnit, Polymerized_2, ..., Polymerized_50] structures, we calculate the Tanimoto similarity and Dice similarity between neighboring structures. We observe that, after a polymerization degree of 16, the structure does not change much in terms of substructures in their MF representation. The identifier and the number of detected substructures are listed for each representation in Figure 4. As expected, more substructures are detected in Polymerized_16 than in monomer and RepeatUnit, with two connection positions "*" (identifier#: 2342113506) are detected in both RepeatUnit and Polymerized_16. When we calculate the RDKit-based descriptors for the list of 51 structures, we observe the convergence around polymerized_16 for such descriptors as well. As an example, the value of the descriptor MaxEStateIndex vs the degree of polymerization is plotted in Figure 4 for reference. We examine the convergence of 20 different polymers that are randomly selected from Data set_1. Polymerized_16 is found to be a long enough oligomer to get converged feature values (see Supporting Information, Section S5 for the convergence curves). In the following parts, we use monomer, RepeatUnit, or Polymerized_16 as our three molecular structure representation of polymers.

3.2. RepeatUnit as Polymer's Structure Representation. The first group of 24 model setups is based on the RepeatUnit structure representation (the top row in Figure 5). Feature representations can be Morgan fingerprint with frequency (MFF), Morgan fingerprint (MF), molecular embedding (ME), or RDKIT-based descriptors. The options for ML algorithms include GPR, LASSO, RF, SVM, FFNN, and CNN. Figure 5 compares the performance of each model in terms of R^2 on Data set_2 (see their predictive R^2, MAE, and RMSE on training set, test set, and Data set_2 in the Supporting Information, Section S6). We particularly place the priority on the model performance on Data set_2 because we are paying more attention to the generalization ability of the obtained ML model. A well-trained model on Data set_1 does not guarantee an excellent transferability to Data set_2. Looking at the GPR model trained with MF, the training/test set R^2 reaches 0.92/0.86—a satisfactory prediction performance, but its R^2 on Data set_2 degrades to 0.55 significantly, although it is also at an acceptable level. The MF-based model generally falls behind using other feature representations, suggesting that the easy-to-use Morgan fingerprint is not necessary for a proper feature representation.

From the perspective of ML algorithms, the best model of each feature representation is indicated with arrows in Figure 5 (see Supporting Information, Section S7 for the detailed parity plot showing the predicted T_g vs experimental T_g or MD simulated T_g). The FFNN model trained on MFF outperforms other models with an R^2 of 0.65 on the Data set_2, and almost the same performance comes from the CNN model based on ME. Both FFNN and CNN models are highly nonlinear models that are supposed to be more powerful in establishing complex correlations, but they require distinct feature representations for better T_g prediction. For example, if changing the feature representation from MFF to MF, the FFNN model's R^2 on Data set_2 changes from 0.65 to 0.50—a decrease of 23%. While changing the feature representation from ME to MFF, the CNN model's R^2 on Data set_2 reduces from 0.65 to 0.54—a decrease of 17%. The interdependence of the feature representation and ML algorithms affects the model's performance significantly.

3.3. Polymerized_16 as Polymer's Structure Representation. The second group of 24 model setups is based on the Polymerized_16 structure representation (the middle row in Figure 5). It is worth noting that the feature extraction here takes more time due to the complexity of Polymerized_16 structures. Figure 4 has demonstrated the descriptor convergence after using polymerized structure, but the expensive calculation becomes an obvious shortcoming compared to other feature engineering methods. It should be noted that most of the Polymerized_16-based models do not outperform the RepeatUnit-based models as shown in Figure 5. The performance of each model is listed in Supporting Information, Section S6, based on which we notice 16/24 RepeatUnit-based models have better performance than the corresponding Polymerized_16-based models. We realize that the polymerized structure representation does not provide much more information than a single repeat unit. The repeat unit structure representation can be considered enough to capture the bonding information and key substructures related to polymer's T_g. Considering Polymerized_16-based models require expensive calculation when processing Polymerized_16 structures and they barely improve model performance compared to RepeatUnit-based models, Polymerized_16 structure representation is found to be inferior to RepeatUnit structure representation.

The best ML models based on Polymerized_16 are mainly GPR and RF (see Supporting Information, Section S7 for their R^2 and parity plots). It is not surprising that GPR and RF models are among the best models. GPR intrinsically searches all accessible functions that best fit the training data; namely, no constraints on the form of the fitting equation are applied. Polymer Genome[8] is a successful application of the GPR model for polymer property predictions, including seven properties such as bandgap, dielectric constant, refractive index, atomization energy, T_g, solubility parameter, and density. Compared to GPR, the RF model uses the advantage of ensemble average for a better prediction. The training/test R^2 from RF models are always higher than those of the other models due to its 100 individual decision trees working together. RF model has demonstrated its excellence in the predictive task of T_g for glasses as well.[84] For our similar noncrystalline polymer materials, the situation is essentially observed to be the same.

3.4. Monomer as Polymer's Structure Representation. The third group of 24 model setups is based on the monomer structure representation (the bottom row in Figure 5). Their performances are generally slightly worse than the previous two groups of models. A R^2 around 0.85 and an MAE around 30 using two-monomer structure representation were reported in the literature for T_g prediction,[43] which is at the same level as our single monomer-based models. Only through the evaluation on Data set_2, it is realized that the model has a relatively poor generalization ability. The use of monomer structure may not be a significant issue when training on a small group of data. Still, it possesses insufficient ability to generalize if examined on a large extra data set. The feature engineering of MFF, ME, and descriptors using monomer can incorporate more structure information than the raw MF and thus improve the model performance to some extent. Still, all the monomer-based ML models are generally not as good as

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

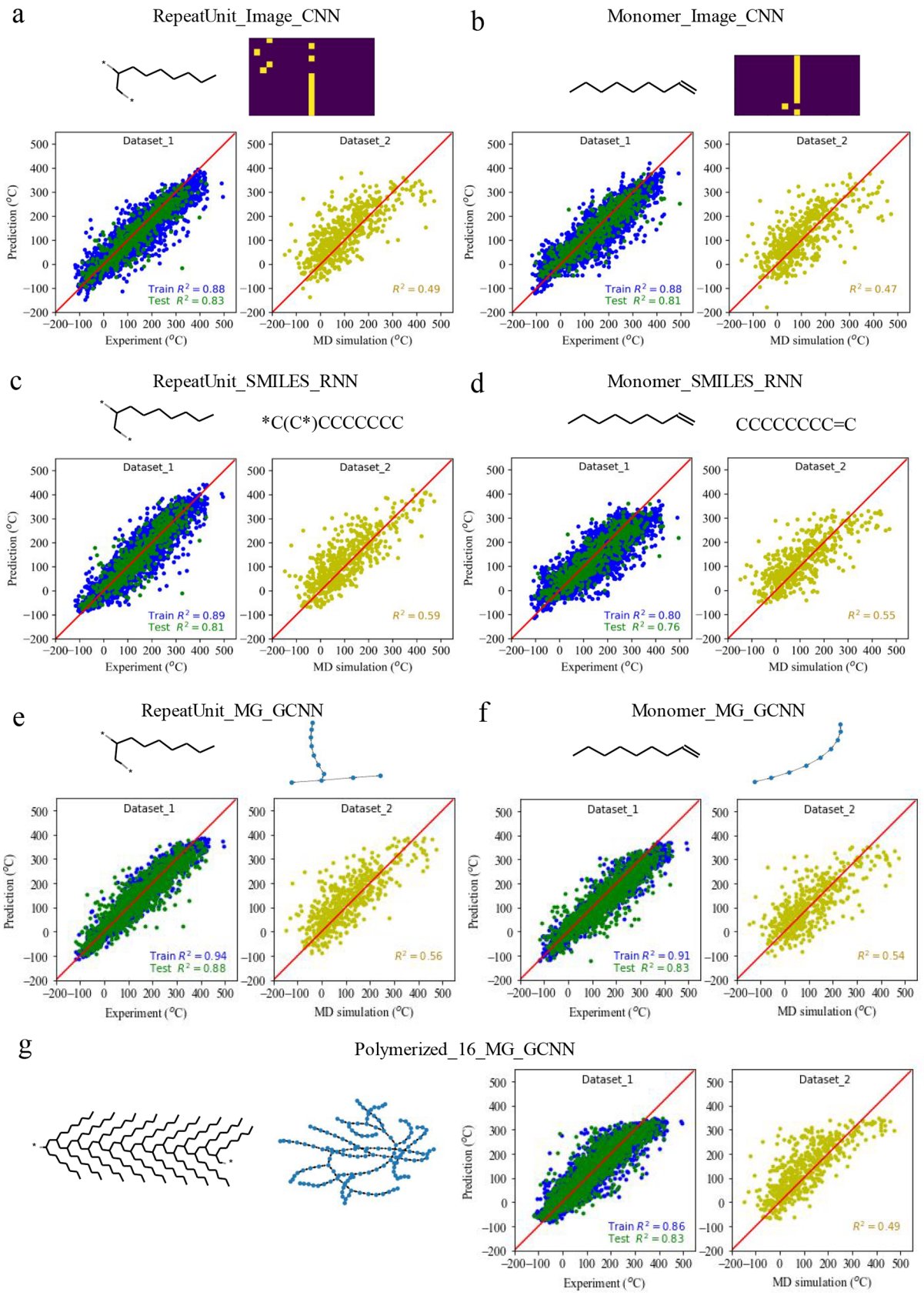


Figure 6. Performance of CNN models, RNN models, and GCNN models. The structure representation and feature input (Image or SMILES) are illustrated on the top of each panel using poly(non-1-ene) as the example. (a) RepeatUnit-based CNN model using 2D Image. (b) Monomer-based CNN model using 2D Image. (c) RepeatUnit-based RNN model using SMILES. (d) Monomer-based RNN model using SMILES. (e) RepeatUnit-based GCNN model using MG. (f) Monomer-based GCNN model using MG. (g) Polymerized_16-based GCNN model using MG.

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

Table 1. Performance of 2D CNN Models, RNN Models, and GCNN Models

		RepeatUnit			Monomer			Polymerized_16		
model	matrix	train	test	Data set_2	train	test	Data set_2	train	test	Data set_2
2D CNN	R2	0.88	0.83	0.49	0.88	0.81	0.47	not applicable		
	MAE	25.60	35.28	60.45	25.60	35.28	62.32			
	RMSE	38.08	48.45	79.22	38.08	48.45	80.96			
RNN	R2	0.89	0.81	0.59	0.80	0.76	0.55	not applicable		
	MAE	26.94	32.21	54.28	36.37	40.05	56.89			
	RMSE	36.92	45.83	71.56	49.01	53.94	74.69			
GCNN	R2	0.94	0.88	0.56	0.91	0.83	0.54	0.86	0.83	0.49
	MAE	18.92	27.10	56.22	24.25	32.46	57.38	30.72	32.88	59.89
	RMSE	25.85	38.82	74.09	32.44	44.40	75.63	41.46	45.02	79.18

the one using single repeat unit or polymerized oligomers, as shown in Figure 5.

The best ML models based on monomer remain to be GPR and RF (see Supporting Information, Section S7 for their R^2 and parity plots), but their R^2 on Data set_2 degrade around 10% compared to the best GPR and RF models using RepeatUnit or Polymerized_16 structures. Monomer is found to be deficient in representing long-chain polymers in the T_g prediction task. One important missing piece of information is the connecting point "*" that is essential for the polymerization process of polymers and is observed to be crucial for more accurate ML models. Without the "*", the monomer's SMILES notation is like that of small organic molecules, leading to the difficulty in differentiating the chemical structures between polymers and small molecules. The other concern of using monomers is that the same monomer structure may polymerize into different polymers through different polymerization pathways. Such one-to-many mapping brings confusion to the ML model and makes it problematic to establish a clear structure−property correlation.[64]

3.5. Special Treatments of Polymer Structure as 2D Image, Character Sequence, or Molecular Graph. 2D CNN model, RNN model, and GCNN model originate from different subjects of image recognition, natural language processing, and graph theory, respectively.[85−87] In the cheminformatics field, small molecular generations[88−94] and property prediction[13,95−98] have used these techniques extensively. We have implemented a 2D CNN model based on images and revealed its poor generalization ability for T_g prediction in the recent study.[44] Besides this, we have also implemented the RNN model that is purely linguistic-based using the SMILES notation of a repeat unit as input.[99] We re-evaluate our previously trained models using the new 566 MD simulations and compare them here with other models. Besides using the RepeatUnit as polymer structure representation, we build new CNN and RNN models based on monomer and compare their performance in Figure 6a−d. The structure representation Polymerized_16 can contain hundreds or thousands of characters in a SMILES notation, making it not applicable here for image processing or SMILES character processing. For example, the maximum input length of the SMILES string is prescribed as 120 (an optimized hyperparameter) for the RNN model.[99] Such a shorter sequence constraint can reduce the training difficulties to get a better model performance. Similarly, the maximum size of the 2D image is prescribed as 310 × 21 so that the obtained 2D matrix is not too sparse while large enough to represent most SMILES. But unfortunately, the long strings of Polymerized_16 can easily break the SMILES length limit for the established RNN and CNN architectures. The requirement on the length of the SMILES is a limitation of RNN and CNN models compared with others. Dimension reduction methods like Principal Component Analysis (PCA), t-SNE, or even Variational Autoencoder (VAE) are possible ways to address the SMILES length issue for RNN and CNN models, but the extra processing converts the SMILES representation to a different feature representation with more complexity. To have a fair comparison for this benchmark study, we do not apply further processing on SMILES representation or other representations, otherwise, it would become another topic to address. Figure 6 illustrates the structure representation and feature input on the top of each panel using poly(non-1-ene) as the example. For the CNN model, the performance on the training set and test set are acceptable using 2D images based on either RepeatUnit or monomer. However, their generalization ability degrades significantly on Data set_2. The RNN model avoids the step of feature engineering, establishing a direct relationship between the SMILES sequence and T_g. Its performance−especially on the Data set_2 is much better than that of 2D CNN. The model performance is summarized in Table 1.

The GCNN model using MG can also be applied to the polymer's T_g prediction. Figure 6e−g compares three GCNN models based on monomer, RepeatUnit, and Polymerized_16, respectively. The structures of poly(non-1-ene) are used as schematic diagrams on the top of each panel. It illustrates the conversion of the atoms and bonds into graph vertices and edges. The more complex a graph is, the more time GCNN needs to train the model. Judging by the obtained R^2, the RepeatUnit-based GCNN model performs slightly better than the monomer-based GCNN model, but unexpectedly a more complex GCNN from the Polymerized_16 structure results in degraded performance. Polymerized_16 is a closer analog to the long-chain polymer than monomer, as demonstrated by the convergence pattern in terms of substructure similarity and descriptor calculation (cf. Figure 4). Intuitively, the Polymerized_16-based model should outperform others, but obviously, the GCNN model here does not favor the use of such complex graph input. This is indicative of the fact that although the Polymerized_16 graph has much higher complexity, it remains void of other large polymeric chain attributes such as excluded volume, chain flexibility, and interchain interactions, all of which play a critical role in determining T_g.

3.6. Summary of 79 ML Models for Polymer's T_g Prediction. We have formulated 79 models using different structure representations, feature representations, and ML algorithms for polymer's T_g prediction. By averaging the R^2 on Data set_2 from different perspectives, we find the ranking of

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

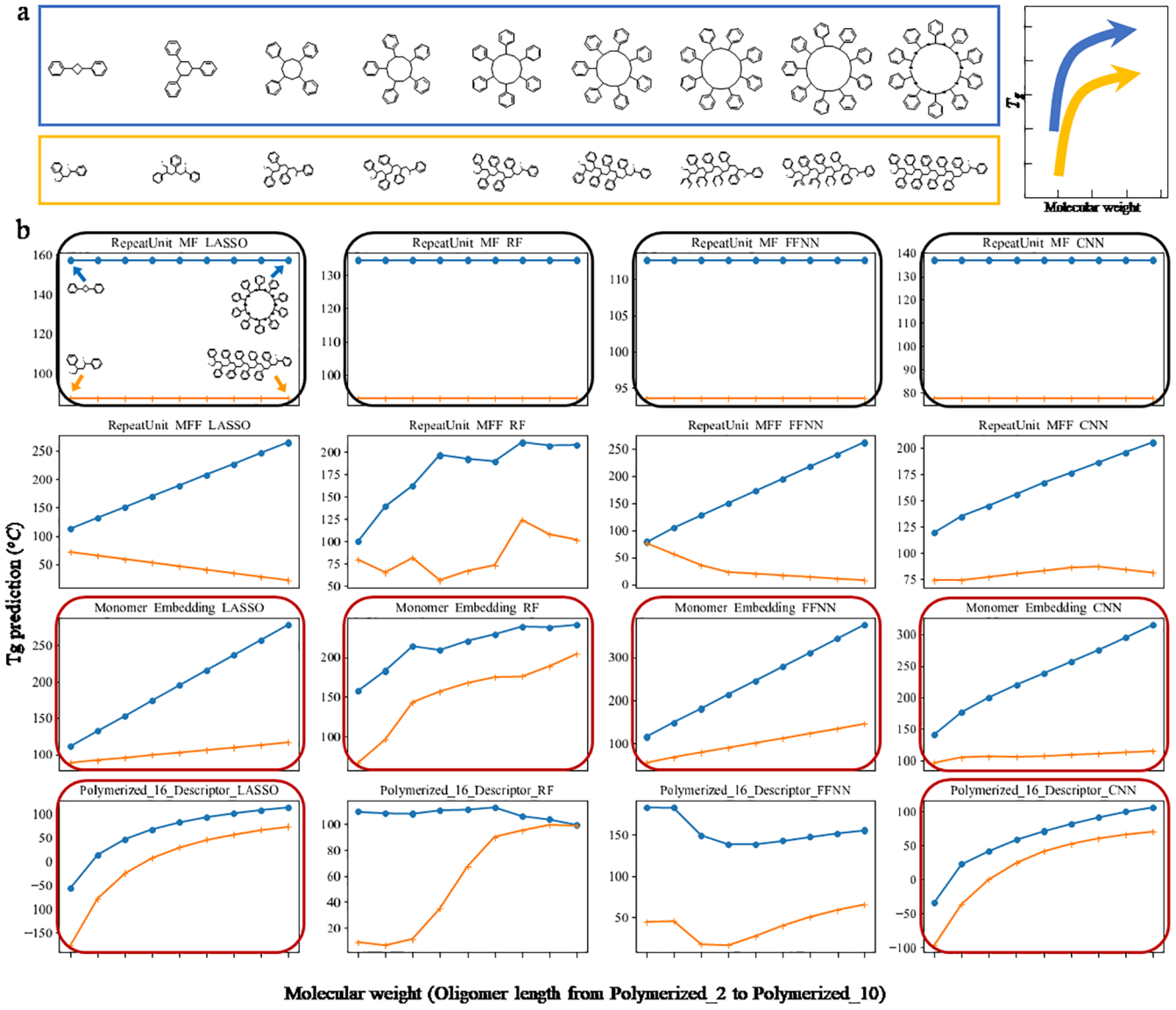


Figure 7. Qualitative examination of T_g prediction on different polystyrene architectures. (a) T_g of polystyrene is a function of molecular weight and topology (cyclic,linear). The experimental T_g trend is based on the study by Grayson et al.[108] (b) We evaluate the T_g of cyclic architecture (blue curves) and linear architecture (orange curves) of polystyrene at different molecular weights by the obtained 79 models. The results of 24 models are displayed here, while the complete results of all 79 models can be found in Supporting Information, Section S8. Red boxes highlight models whose predictions match with the experimental trend qualitatively. Black boxes highlight models that are completely insensitive to topology and molecular weight.

the structure representation from the best to the worst is RepeatUnit > Polymerized_16 > Monomer. We should emphasize that the Polymerized_16 structures are not superior to RepeatUnit_1, and using a single repeat unit is good enough to incorporate key substructures related to polymer's T_g. Monomer contains less substructure information and ignores the polymerization information "*". Besides, the monomer structure cannot map to a unique polymer. Thus, it is not surprising that Monomer is the worst structure representation for polymer informatics.

The ranking of the feature representation from the best to the worst is MFF > SMILES > Descriptor > ME > MG > MF > 2D Image. The poor performance of using 2D images suggests that converting a physical problem into an image-processing problem is not a good idea, as we discussed before.[44] When structural features are implicitly represented by an image, it is difficult for an ML algorithm to reveal a reliable structure−property relationship. MF proves to be a poor feature representation too, showing that only marking the presence or absence of each substructure is not sufficient, particular, for the T_g prediction.[44] MFF, however, considers the occurring frequency of each substructure and turns out to be the best feature representation. Note that from the classical group contribution theory,[24] the polymer's T_g can be reasonably predicted by considering the contributions from different chemical groups.[100−102] Similarly, our MFF feature representation considers the chemical substructures and their occurring frequencies of polymers, leading to a better prediction on T_g.

The ranking of the ML algorithms from the best to the worst is RF > CNN > RNN > GPR > FFNN > GCNN > LASSO >

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

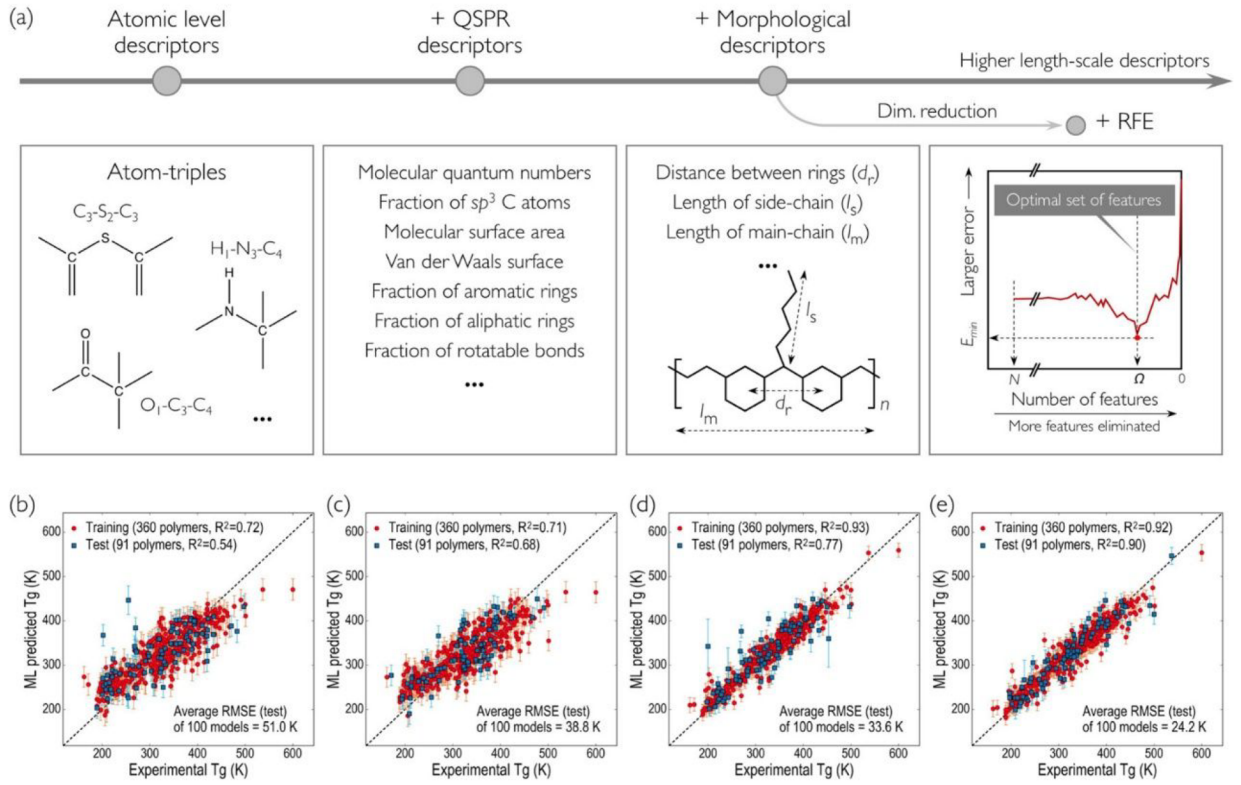


Figure 8. Three-level hierarchical descriptors used by the Polymer Genome platform, and the model performance on T_g prediction considering different types of fingerprints. (a) Three levels of hierarchical descriptors: atomic level descriptors, QSPR descriptors, and morphological descriptors. Fingerprint dimensions are reduced by a recursive feature elimination (RFE) process. (b–d) Model performance improves by using only atomic level descriptors, atomic level and QSPR descriptors, entire fingerprint components including morphological descriptors, and an RFE-processed fingerprint. The figure is reprinted with permission from ref 8. Copyright 2018 American Chemical Society.

SVM > 2D CNN. The first six algorithms are highly nonlinear ML models that are supposed to be more powerful than the linear regression method, such as LASSO. RF model owns its excellent performance to its ensemble attribution, and the improved predictive accuracy outperforms all the other models. A surprisingly good model is the RNN model that only reads the SMILES and does not need any feature processing. It demonstrates that the SMILES sequence may contain several of the essential structural features (such as different types of atoms and how they are topologically connected) related to polymer's T_g. The 2D CNN model performs the worst due to the poor feature representation via 2D images, but 1D CNN models based on other feature representations are better than most ML algorithms. There is a strong interdependence of the feature representation and ML algorithms to affect the model's accuracy and generalization ability significantly.

Among all the structure representation, feature representation, and ML algorithms, the best model setup is found to be the RepeatUnit-based CNN model using ME. Its predictive performance has a training/test R^2 of 0.84/0.82 on Data set_1 and an R^2 of 0.65 on Data set_2. The worst model setup is the monomer-based SVM model using ME. Its predictive performance has a training/test R^2 of 0.79/0.74 on Data set_1 and an R^2 of 0.44 on Data set_2. The best model improves the predictive accuracy on Data set_2 by 47.73% compared to the worst model. A proper ML model setup makes a big difference in a certain task. Similar to other benchmark studies for specific problems like molecule property predicitons,[103−105] images classifications,[106] or text processing,[107] etc., our benchmark results are specifically for polymer's T_g prediction, but the revealed pros and cons behind different models are of great interests for other polymer informatics tasks.

3.7. Sensitivity of the ML Model to Topology and Molecular Weight of Polymer Chains. We train all the ML models on the Data set_1 of homopolymers and then evaluate their performances using the unlabeled homopolymers in Data set_2. The model transferability from one data set to another one has been carefully examined, but no topology or molecular weight information has been explicitly taken into account during model training. However, an experimental study of polystyrene has demonstrated that the polymer's T_g depends on the topology and molecular weight of polymer chains.[108] The polystyrene's T_g dependence on molecular weight is also empirically defined with Flory–Fox equation.[109] When polystyrene is in the form of a cyclic chain (ring polymer), its T_g measurement is higher than its linear compartment.[108] When polystyrene possesses higher molecular weight, its T_g also increases accordingly. Such a qualitative trend is illustrated in Figure 7a, where a longer chain polystyrene (with higher molecular weight) is supposed to have a higher T_g than a short-chain, and a cyclic chain is supposed to have a higher T_g than its linear analog. Therefore, we use the polystyrene case as a qualitative examination to check if our obtained 79 models can handle the T_g dependence on topology and molecular weight of polymer chains.

To consider molecular weight of polymer chain, we use oligomers with increased chain length from Polymerized_2 to

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

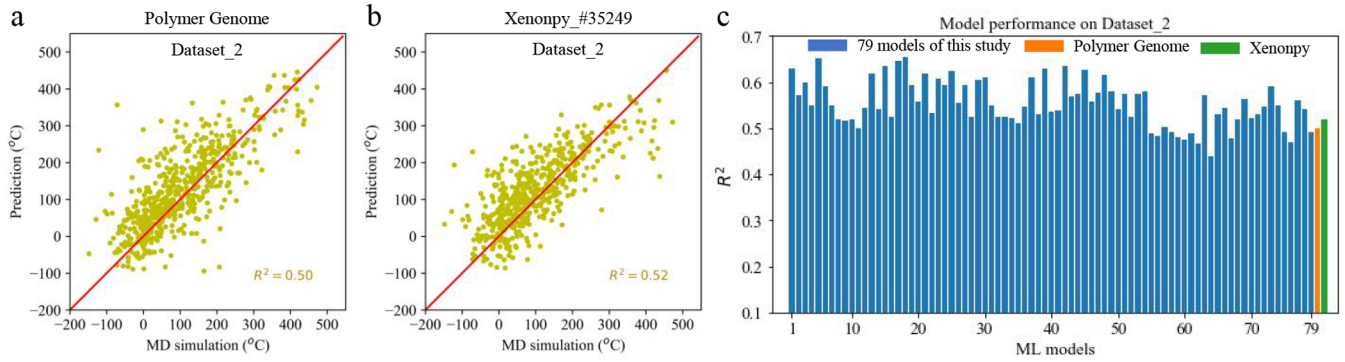


Figure 9. Model performance on Data set_2. (a) Parity plot obtained with the Polymer Genome platform. (b) Parity plot obtained with the Xenonpy model. (c) Performance comparison of 79 models of this study, the Polymer Genome platform (orange) and the Xenonpy (green) model.

Polymerized_10 as shown in Figure 7a for polystyrene. Correspondingly, a longer SMILES contains more atoms to be processed with feature engineering, taking into account a higher molecular weight. The cyclic topological information is considered in the SMILES by concatenating the structure's head and tail location indicated by the "*" symbol. Figure 7a shows that the linear topologies are open-end chain structures with head and tail "*", while the cyclic topologies are cycle-like structures with head and tail being concatenated. It is worth noticing that our obtained 79 models are not trained for handling different molecular weight or topology of polymer chains. They are trained using either monomer, RepeatUnit, or Polymerized_16 structure representations, and they are not supposed to take a longer or cyclic structure representation for T_g prediction. However, if a previous trained model is given a longer or a cyclic chain, it is interesting to check whether the model can at least qualitatively predict a higher T_g value. Such a sensitivity check is only an extra examination of these obtained models as an additional merit.

As long as the pattern of the obtained T_g prediction for a longer or cyclic polymer chain matches with the experimental observation, a model is considered as sensitive to topology and molecular weight. Some qualified models are highlight by the red boxes in Figure 7b. 14/79 models are found to comply with the experimental pattern qualitatively, and the obtained best model—RepeatUnit-based CNN model using ME—is one of them. Another consistent observation is that ML models using MF feature representation are entirely insensitive to topology and molecular weight (highlighted by the black boxes in the first row of Figure 7b), as it does not consider the occurring frequency of substructures. Whether the input structure is cyclic or linear, and no matter the molecular weight of the input structure is high or low, MF always identifies the same presence (on)/absence (off) condition for most substructures. The topology and molecular weight differences cannot be recognized by MF feature engineering. Therefore, ML models using MF generate the same T_g value and are not able to match with the experimental T_g trend in Figure 7a. On the contrary, ML models using MFF feature representation are sensitivity to topology and molecular weight (the second row of Figure 7b), although the trends are not consistent with the monotonically increasing patterns in experiments. MF feature engineering is demonstrated missing important informations compared to MFF feature engineering, or other feature engineerings like embedding or descriptor shown in the third and fourth row of Figure 7b. The results for all 79 models can be found in the Supporting Information, Section S8.

3.8. Comparison between our ML Models vs Other Successful Ones. One successful platform for polymer's T_g prediction is the Polymer Genome developed by Ramprasad and co-workers.[8] Figure 8 shows the three levels of hierarchical descriptors used in the Polymer Genome platform: atomic level descriptors, QSPR descriptors, and morphological descriptors. For the atomic level descriptors, the occurrence of a fixed set of fragments or motifs is tracked, such as one-fold coordinated oxygen and 3-fold coordinated carbon. The QSPR descriptors are similar to that of molecular descriptors in RDKit, including the van der Waals surface area, the topological polar surface area, the fraction of rotatable bonds, etc. The morphological descriptors are designed to represent the polymer chain features, such as the shortest topological distance between rings, fraction of atoms that are part of side-chains, and the length of the largest side-chain. Overall, there are 953 components for each polymer's fingerprint vector, including 371 atomic level descriptors, 522 QSPR descriptors, and 60 morphological descriptors. More details about the three levels of hierarchical descriptors are given in the Polymer Genome platform.[102] Based on the three levels of hierarchical descriptors, their optimized GPR model shown in Figure 8e has a training/test R^2 of 0.92/0.90—a satisfactory accuracy on their data sets of 471 polymers (360 polymers for training and 91 polymers for test). We query the T_g prediction using the Polymer Genome platform for our 566 MD simulated polymers and obtain an R^2 of 0.50, shown in Figure 9a. Figure 9c compares the performance of our 79 models with the Polymer Genome platform, and a ~30% difference in terms R^2 of these models is observed. Overall, the predictions from the Polymer Genome platform are comparable with our best ML models, considering the smaller train/test (360/91) data sets of this model.

The other successful platform for polymer's property prediction is Xenonpy, developed by Yoshida and co-workers.[110] They provide more than 140 000 pretrained neural networks for researchers to carry out neural transfer, learning from 12 properties of 133 805 small organic molecules in the QM9 data set.[111−114] First, a fully connected pyramid neural network is trained using training instances from the monomeric properties. Afterward, a subnetwork other than the output layer is used as a feature extractor. It is repurposed on a model of the polymer's property, such as T_g values from 5917 unique homopolymers in PoLyInfo.[25] On polymer's T_g prediction, there are 200 models pretrained and easy-to-use via

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

their server API. We download one of their best models #35249, which has three hidden layers and bases on 2048 bits mixed fingerprints including RDKit fingperprint, ECFP, MACCS, etc. Its training R^2 reaches to 0.92 on their data set. Similarly, we make T_g predictions using the Xenonpy model for our 566 MD simulated structures. Its best predictive ability reaches to an R^2 of 0.52 after we retrain the model using our data set based on MF, shown in Figure 9b. Figure 9c demonstrates a comparable performance for the Xenonpy model and other models. Comparing our obtained model to these two successful platforms, we find that their carefully tuned models can generate reasonable predictions on the T_g values of unlabeled Data set_2. However, further optimizations are still possible if using different structure representations, feature representation, or ML algorithms.

The comparisons between the Polymer Genome model, the Xenonpy model, and our ML models demonstrate their good generalization abilities on new structures, using the true T_g values from MD simulations. However, using new structures with experimental T_g is more desired to verify the generalization ability of ML models. Therefore, we collected an experimental database of conjugated polymers.[27,115] Conjugated polymers possess promising optical and electronic properties, and their aromatic backbone and alkyl side chain chemistry differs drastically. There are 62 conjugated polymers that are new to our Data set_1 and Data set_2, which is an ideal experimental data set to examine these ML models. Simillary, our models demonstrate a comparable performance as Polymer Genome models and Xenonpy models (see Supporting Information, Section S9, for detailed results). Thus, the two successful platforms and our ML models are demonstrated able to generalize on a certain class of polymer.

4. CONCLUSION

Polymer's T_g prediction is a vital polymer informatics task that requires a combined knowledge of polymer structures, feature engineering, and ML algorithms. Using the right polymer structure representations, generating suitable feature representations, and implementing proper ML algorithms are key steps to formulate a reliable ML model with satisfying accuracy and generalization ability. Here we carry out a systematic benchmark study to investigate the performance of different model setups, using our collected large data sets of homopolymers. The model training process involves 5-fold cross-validation and test set evaluation using 6923 homopolymers in Data set_1. As we focus more on the generalization ability of the obtained model, our most crucial evaluation metric of the model performance is the predictive R^2 on the MD simulated 566 structures from the unlabeled Data set_2. We investigate three structure representations like monomer, RepeatUnit, or Polymerized_16. Based on each structure representation, we consider seven feature representations such as MFF, MF, ME, Descriptors, SMILES, Image, and MG. Then we implemented eight ML algorithms, including GPR, LASSO, RF, SVM, FFNN, RNN, CNN, and GCNN. In total, we develop 79 models to investigate the pros and cons behind different model setups.

Based on our obtained results, important findings and observations are as follows. (1) Polymerized_16—the oligomer corresponding to 16 repeat units chained together—is a long enough analog to represent long-chain polymer due to it is convergence in terms of substructures and descriptors. It retains the bonding information "*" and most substructures as in the original long-chain structure, but some models do not prefer the use of polymerized structure over a single repeat unit. In most circumstances, using one repeat unit is sufficient to capture the main structure information related to polymer's T_g. Furthermore, the monomer structure contains much less substructure information and ignores the bonding information between adjacent repeating units completely. We consistently observe that the monomer-based models are not as good as those using single repeat or polymerized oligomers. We find the ranking of the structure representation from the best to the worst is RepeatUnit > Polymerized_16 > monomer. (2) The ranking of the feature representation from the best to the worst is MFF > SMILES > Descriptor > ME > MG > MF > 2D Image. The easy-to-use Morgan fingerprint that only marks the existence (on)/absence (off) of each substructure is not necessary as a suitable feature representation compared to other options. On the contrary, MFF, which considers the frequency of occurrence for each substructure, turns out to be the best feature representation. Moreover, it is worth noticing that 2D images prove to be not ideal as a feature representation of polymers, as the patterns of 2D images cannot retain important chemical structural information on polymers. (3) Based on the average model performance, the ranking of the ML methods for T_g prediction from the best to the worst is RF > CNN > RNN > GPR > FFNN > GCNN > LASSO > SVM > 2D CNN. Thanks to the ensemble attribution, the RF algorithm demonstrates excellent performance, while GPR uses its nonparametric approach and kernel trick to be among the best models. CNN, RNN, and FFNN are highly nonlinear models that are more powerful than linear models like LASSO and SVM. Although GCNN has an advantage of learnable featurizations,[35,103,116,117] its performance for T_g prediction is not demonstrated to be superior to others. (4) A high training/test R^2 on Data set_1 does not necessarily guarantee a good generalization ability to the unlabeled Data set_2. Thus, models that have good transferability to Data set_2 are considered better ones. Among our formulated 79 models, the best model is the RepeatUnit-based CNN model using ME. Its predictive performance has a training/test R^2 of 0.84/0.82 on Data set_1 and an R^2 of 0.65 on Data set_2. All models' sensitivity to topology and molecular weight are checked qualitatively using the cyclic/linear structure of polystyrene. Fourteen of 79 models comply with the experimental trend, and the obtained best model—the RepeatUnit-based CNN model using ME—is also among them. When compared with successful platforms like Polymer Genome and Xenonpy, our models demonstrate a comparable performance based on Data set_2 or an experimental database of conjugated polymers. Good generalization abilities are observed in our formulated models.

In summary, our benchmark study investigates the synergy of structure representations, feature representations, and ML algorithms on the polymer's T_g prediction, by taking advantage of large and diverse data sets and high-throughput MD simulations for model training and validation. The revealed pros and cons behind different model setups provide useful guidance to better address the polymer's T_g prediction task and also a good reference for other polymer informatics tasks.

Data and Software Availability. Homopolymers and their corresponding T_g values and SMILES notations were collected from PolyInfo, which is freely available (https://polymer.nims.go.jp/en/). Machine learning models are freely available on GitHub (https://github.com/figotj/Tg_Benchmarking). CSV

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

files containing MD simulated polymers along with their SMILES and predicted T_g are provided in the Supporting Information. The MD simulations are carried out using the open source program LAMMPS (https://www.lammps.org/). The machine learning models are built with open source python libraries Tensorflow (https://www.tensorflow.org/) and Scikit-learn (https://scikit-learn.org/stable/). The package versions are provided on GitHub (https://github.com/figotj/Tg_Benchmarking).

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c01031.

Pairwise structural similarity characterized by Dice coefficient, feature engineering methods, model parameters in ML algorithms, model performance versus training set size, convergence of polymerization degree of 20 randomly selected polymers in terms of substructures and descriptors, performance of ML models in terms of R^2, MAE and RMSE, parity plots of different models showing the predictive T_g vs experimental T_g or MD simulated T_g, sensitivity of the model to topology and molecular weight, snd ML predicted T_g vs experimental T_g for 62 conjugated polymers in a newly reported experimental study (PDF)

Data for 566 MD simulated polymers and their T_g values (XLSX)

Predictions from 79 ML models (XLSX)

■ AUTHOR INFORMATION

Corresponding Author

Ying Li − *Department of Mechanical Engineering and Polymer Program, Institute of Materials Science, University of Connecticut, Storrs, Connecticut 06269, United States;* orcid.org/0000-0002-1487-3350; Phone: (860) 483-7110; Email: ying.3.li@uconn.edu; Fax: (860) 486-5088

Authors

Lei Tao − *Department of Mechanical Engineering, University of Connecticut, Storrs, Connecticut 06269, United States*

Vikas Varshney − *Materials and Manufacturing Directorate, Air Force Research Laboratory, Wright Patterson Air Force Base, Ohio 45433, United States;* orcid.org/0000-0002-2613-458X

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c01031

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We gratefully acknowledge financial support from the Air Force Office of Scientific Research through the Air Force's Young Investigator Research Program (FA9550-20-1-0183; Program Manager: Dr. Ming-Jen Pan) and the National Science Foundation (CMMI-1934829). Y.L. would like to express thanks for the support from 3M's Non-Tenured Faculty Award. This research also benefited in part from the computational resources and staff contributions provided by the Booth Engineering Center for Advanced Technology (BECAT) at UConn. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Department of Defense. The authors also acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin (Frontera project and National Science Foundation Award 1818253) for providing HPC resources that have contributed to the research results reported within this paper. We are indebted to generous helps from Dr. Stephen Wu (The Institute of Statistical Mathematics, Tokyo, Japan) for the machine learning models in XenonPy, and Drs. Rampi Ramprasad and Chiho Kim at Georgia Tech for providing access to the Polymer Genome platform.

■ NOMENCLATURE

symbol	meaning
ML	machine learning
MD	molecular dynamics
Monomer	monomer structure before polymerization
RepeatUnit	single repeat unit structure
Polymerized_N	oligomer with N repeat units chained together
SMILES	simplified molecular input line entry system
MFF	Morgan fingerprint considering substructures' frequency of occurrence
MF	Morgan fingerprint
ME	molecular embedding
MG	molecular graph
LASSO	least absolute shrinkage and selection operator
SVM	support vector machine
FFNN	feed-forward neural networks
CNN	convolutional neural networks
GCNN	graph convolutional neural networks
RNN	recurrent neural networks
RF	random forests
GPR	Gaussian process regression

■ REFERENCES

(1) Chen, L.; Pilania, G.; Batra, R.; Huan, T. D.; Kim, C.; Kuenneth, C.; Ramprasad, R. Polymer informatics: Current status and critical next steps. *Mater. Sci. Eng., R* **2021**, *144*, 100595.

(2) Audus, D. J.; de Pablo, J. J. Polymer informatics: Opportunities and challenges. *ACS Macro Lett.* **2017**, *6*, 1078−1082.

(3) Sha, W.; Li, Y.; Tang, S.; Tian, J.; Zhao, Y.; Guo, Y.; Zhang, W.; Zhang, X.; Lu, S.; Cao, Y. C.; et al. Machine learning in polymer informatics. *InfoMat* **2021**, *3*, 353−361.

(4) Wu, S.; Yamada, H.; Hayashi, Y.; Zamengo, M.; Yoshida, R., Potentials and challenges of polymer informatics: exploiting machine learning for polymer design. *arXiv preprint* 2020; arXiv:2010.07683.

(5) Chen, G.; Shen, Z.; Iyer, A.; Ghumman, U. F.; Tang, S.; Bi, J.; Chen, W.; Li, Y. Machine-Learning-Assisted De Novo Design of Organic Molecules and Polymers: Opportunities and Challenges. *Polymers* **2020**, *12*, 163.

(6) Gormley, A. J.; Webb, M. A. Machine learning in combinatorial polymer chemistry. *Nat. Rev. Mater.* **2021**, *6*, 642−644.

(7) Patra, A.; Batra, R.; Chandrasekaran, A.; Kim, C.; Huan, T. D.; Ramprasad, R. A multi-fidelity information-fusion approach to machine learn and predict polymer bandgap. *Comput. Mater. Sci.* **2020**, *172*, 109286.

(8) Kim, C.; Chandrasekaran, A.; Huan, T. D.; Das, D.; Ramprasad, R. Polymer genome: a data-powered polymer informatics platform for property predictions. *J. Phys. Chem. C* **2018**, *122*, 17575−17585.

(9) Chen, L.; Kim, C.; Batra, R.; Lightstone, J. P.; Wu, C.; Li, Z.; Deshmukh, A. A.; Wang, Y.; Tran, H. D.; Vashishta, P.; et al.

Frequency-dependent dielectric constant prediction of polymers using machine learning. *npj Comput. Mater.* **2020**, *6*, 1−9.
(10) Lightstone, J. P.; Chen, L.; Kim, C.; Batra, R.; Ramprasad, R. Refractive index prediction models for polymers using machine learning. *J. Appl. Phys.* **2020**, *127*, 215105.
(11) Wu, S.; Kondo, Y.; Kakimoto, M.-a.; Yang, B.; Yamada, H.; Kuwajima, I.; Lambard, G.; Hongo, K.; Xu, Y.; Shiomi, J.; et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *npj Comput. Mater.* **2019**, *5*, 1−11.
(12) Miccio, L. A.; Schwartz, G. A. From chemical structure to quantitative polymer properties prediction through convolutional neural networks. *Polymer* **2020**, *193*, 122341.
(13) Miccio, L. A.; Schwartz, G. A. Localizing and quantifying the intra-monomer contributions to the glass transition temperature using artificial neural networks. *Polymer* **2020**, *203*, 122786.
(14) Ning, L. Artificial neural network prediction of glass transition temperature of fluorine-containing polybenzoxazoles. *J. Mater. Sci.* **2009**, *44*, 3156−3164.
(15) Liu, W. Prediction of glass transition temperatures of aromatic heterocyclic polyimides using an ANN model. *Polym. Eng. Sci.* **2010**, *50*, 1547−1557.
(16) Palomba, D.; Vazquez, G. E.; Díaz, M. F. Novel descriptors from main and side chains of high-molecular-weight polymers applied to prediction of glass transition temperatures. *J. Mol. Graphics Modell.* **2012**, *38*, 137−147.
(17) Mattioni, B. E.; Jurs, P. C. Prediction of glass transition temperatures from monomer and repeat unit structure using computational neural networks. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 232−240.
(18) Liu, W.; Cao, C. Artificial neural network prediction of glass transition temperature of polymers. *Colloid Polym. Sci.* **2009**, *287*, 811−818.
(19) Pei, J. F.; Cai, C. Z.; Zhu, Y. M.; Yan, B. Modeling and Predicting the Glass Transition Temperature of Polymethacrylates Based on Quantum Chemical Descriptors by Using Hybrid PSO-SVR. *Macromol. Theory Simul.* **2013**, *22*, 52−60.
(20) Higuchi, C.; Horvath, D.; Marcou, G.; Yoshizawa, K.; Varnek, A. Prediction of the Glass-Transition Temperatures of Linear Homo/Heteropolymers and Cross-Linked Epoxy Resins. *ACS Appl. Polym. Mater.* **2019**, *1*, 1430−1442.
(21) Pilania, G.; Iverson, C. N.; Lookman, T.; Marrone, B. L. Machine-Learning-Based Predictive Modeling of Glass Transition Temperatures: A Case of Polyhydroxyalkanoate Homopolymers and Copolymers. *J. Chem. Inf. Model.* **2019**, *59*, 5013−5025.
(22) Goswami, S.; Ghosh, R.; Neog, A.; Das, B. Deep learning based approach for prediction of glass transition temperature in polymers. *Mater. Today: Proc.* **2021**, *46*, 5838−5843.
(23) Miccio, L. A.; Schwartz, G. A. Mapping Chemical Structure−Glass Transition Temperature Relationship through Artificial Intelligence. *Macromolecules* **2021**, *54*, 1811−1817.
(24) Van Krevelen, D. W.; Te Nijenhuis, K. *Properties of polymers: their correlation with chemical structure; their numerical estimation and prediction from additive group contributions*; Elsevier: 2009.
(25) Otsuka, S.; Kuwajima, I.; Hosoya, J.; Xu, Y.; Yamazaki, M. PoLyInfo: Polymer Database for Polymeric Materials Design. *2011 International Conference on Emerging Intelligent Data and Web Technologies (EIDWT)* **2011**, 22−29.
(26) Tchoua, R. B.; Chard, K.; Audus, D. J.; Ward, L. T.; Lequieu, J.; De Pablo, J. J.; Foster, I. T. Towards a Hybrid Human-Computer Scientific Information Extraction Pipeline. *2017 IEEE 13th International Conference on e-Science (e-Science)* **2017**, 109−118.
(27) Xie, R.; Weisen, A. R.; Lee, Y.; Aplan, M. A.; Fenton, A. M.; Masucci, A. E.; Kempe, F.; Sommer, M.; Pester, C. W.; Colby, R. H.; et al. Glass transition temperature from the chemical structure of conjugated polymers. *Nat. Commun.* **2020**, *11*, 1−8.
(28) Bicerano, J. *Prediction of polymer properties*; CRC Press: 2002.
(29) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.
(30) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.
(31) Weininger, D. SMILES. 3. DEPICT. Graphical depiction of chemical structures. *J. Chem. Inf. Model.* **1990**, *30*, 237−243.
(32) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(33) Todeschini, R.; Consonni, V. *Handbook of molecular descriptors*; John Wiley & Sons: 2008; Vol. *11*.
(34) Jaeger, S.; Fulle, S.; Turk, S. Mol2vec: unsupervised machine learning approach with chemical intuition. *J. Chem. Inf. Model.* **2018**, *58*, 27−35.
(35) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595−608.
(36) Wills, T. J.; Polshakov, D. A.; Robinson, M. C.; Lee, A. A. Impact of Chemist-In-The-Loop Molecular Representations on Machine Learning Outcomes. *J. Chem. Inf. Model.* **2020**, *60*, 4449−4456.
(37) Alkharusi, H. Categorical variables in regression analysis: A comparison of dummy and effect coding. *Int. J. Edu.* **2012**, *4*, 202.
(38) Jha, A.; Chandrasekaran, A.; Kim, C.; Ramprasad, R. Impact of dataset uncertainties on machine learning model predictions: the example of polymer glass transition temperatures. *Modell. Simul. Mater. Sci. Eng.* **2019**, *27*, 024002.
(39) Ramprasad, M.; Kim, C., Assessing and improving machine learning model predictions of polymer glass transition temperatures. *arXiv preprint* 2019; arXiv:1908.02398.
(40) Doan Tran, H.; Kim, C.; Chen, L.; Chandrasekaran, A.; Batra, R.; Venkatram, S.; Kamal, D.; Lightstone, J. P.; Gurnani, R.; Shetty, P.; et al. Machine-learning predictions of polymer properties with Polymer Genome. *J. Appl. Phys.* **2020**, *128*, 171104.
(41) Yu, X. Support vector machine-based QSPR for the prediction of glass transition temperatures of polymers. *Fibers Polym.* **2010**, *11*, 757−766.
(42) Nazarova, A. L.; Yang, L.; Liu, K.; Mishra, A.; Kalia, R. K.; Nomura, K.-i.; Nakano, A.; Vashishta, P.; Rajak, P. Dielectric Polymer Property Prediction Using Recurrent Neural Networks with Optimizations. *J. Chem. Inf. Model.* **2021**, *61*, 2175.
(43) Ma, R.; Liu, Z.; Zhang, Q.; Liu, Z.; Luo, T. Evaluating Polymer Representations via Quantifying Structure−Property Relationships. *J. Chem. Inf. Model.* **2019**, *59*, 3110−3119.
(44) Tao, L.; Chen, G.; Li, Y. Machine learning discovery of high-temperature polymers. *Patterns* **2021**, *2*, 100225.
(45) Kuenneth, C.; Rajan, A. C.; Tran, H.; Chen, L.; Kim, C.; Ramprasad, R. Polymer informatics with multi-task learning. *Patterns* **2021**, *2*, 100238.
(46) Ma, R.; Luo, T. PI1M: A Benchmark Database for Polymer Informatics. *J. Chem. Inf. Model.* **2020**, *60*, 4684−4690.
(47) Schustik, S. A.; Cravero, F.; Ponzoni, I.; Díaz, M. F. Polymer informatics: Expert-in-the-loop in QSPR modeling of refractive index. *Comput. Mater. Sci.* **2021**, *194*, 110460.
(48) Cravero, F.; Schustik, S. A.; Martínez, M. J.; Vázquez, G. E.; Díaz, M. F.; Ponzoni, I. Feature Selection for Polymer Informatics: Evaluating Scalability and Robustness of the FS4RVDD Algorithm Using Synthetic Polydisperse Data Sets. *J. Chem. Inf. Model.* **2020**, *60*, 592−603.
(49) Zhang, Y.; Xu, X. Machine learning glass transition temperature of polymethacrylates. *Mol. Cryst. Liq. Cryst.* **2021**, 1−14.
(50) Zhang, Y.; Xu, X. Machine learning glass transition temperature of polyacrylamides using quantum chemical descriptors. *Polym. Chem.* **2021**, *12*, 843−851.
(51) Bajusz, D.; Rácz, A.; Héberger, K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminf.* **2015**, *7*, 1−13.

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

(52) Landrum, G. *RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling*; Academic Press: 2013.
(53) Zenobi, G.; Cunningham, P. Using Diversity in Preparing Ensembles of Classifiers Based on Different Feature Subsets to Minimize Generalization Error. *Machine Learning: ECML 2001* **2001**, *2167*, 576−587.
(54) Mauri, A. alvaDesc: A tool to calculate and analyze molecular descriptors and fingerprints. *Ecotoxicological QSARs* **2020**, 801−820.
(55) Lin, T.-S.; Coley, C. W.; Mochigase, H.; Beech, H. K.; Wang, W.; Wang, Z.; Woods, E.; Craig, S. L.; Johnson, J. A.; Kalow, J. A.; et al. BigSMILES: a structurally-based line notation for describing macromolecules. *ACS Cent. Sci.* **2019**, *5*, 1523−1531.
(56) Gulli, A.; Pal, S. *Deep learning with Keras*; Packt Publishing Ltd.: 2017.
(57) Chen, G.; Tao, L.; Li, Y. Predicting Polymers' Glass Transition Temperature by a Chemical Language Processing Model. *Polymers* **2021**, *13*, 1898.
(58) O'Shea, K.; Nash, R., An introduction to convolutional neural networks. *arXiv preprint* 2015; arXiv:1511.08458.
(59) Chollet, F., *keras*; 2015.
(60) Pope, P.; Kolouri, S.; Rostrami, M.; Martin, C.; Hoffmann, H., Discovering molecular functional groups using graph convolutional neural networks. *arXiv preprint* 2018; arXiv:1812.00265.
(61) Ramsundar, B.; Eastman, P.; Walters, P.; Pande, V. *Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more*; O'Reilly Media, Inc.: 2019.
(62) Fonti, V.; Belitser, E. Feature selection using lasso. *VU Amsterdam Research Paper in Business Analytics* **2017**, *30*, 1−25.
(63) Muthukrishnan, R.; Rohini, R. LASSO: A feature selection technique in predictive modeling for machine learning. *2016 IEEE International Conference on Advances in Computer Applications (ICACA)* **2016**, 18−20.
(64) Chen, G.; Shen, Z.; Li, Y. A machine-learning-assisted study of the permeability of small drug-like molecules across lipid membranes. *Phys. Chem. Chem. Phys.* **2020**, *22*, 19687−19696.
(65) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(66) Binder, K. *Monte Carlo and molecular dynamics simulations in polymer science*; Oxford University Press: 1995.
(67) Abbott, L. J.; Hart, K. E.; Colina, C. M. Polymatic: a generalized simulated polymerization algorithm for amorphous polymers. *Theor. Chem. Acc.* **2013**, *132*, 1334.
(68) Sun, H.; Mumby, S. J.; Maple, J. R.; Hagler, A. T. An ab initio CFF93 all-atom force field for polycarbonates. *J. Am. Chem. Soc.* **1994**, *116*, 2978−2987.
(69) Sun, H.; Ren, P.; Fried, J. The COMPASS force field: parameterization and validation for phosphazenes. *Comput. Theor. Polym. Sci.* **1998**, *8*, 229−246.
(70) Sun, H. Ab initio calculations and force field development for computer simulation of polysilanes. *Macromolecules* **1995**, *28*, 701−712.
(71) Heinz, H.; Lin, T.-J.; Kishore Mishra, R.; Emami, F. S. Thermodynamically consistent force fields for the assembly of inorganic, organic, and biological nanostructures: the INTERFACE force field. *Langmuir* **2013**, *29*, 1754−1765.
(72) Sun, H.; Ren, P.; Fried, J. The COMPASS force field: parameterization and validation for phosphazenes. *Comput. Theor. Polym. Sci.* **1998**, *8*, 229−246.
(73) Bunte, S. W.; Sun, H. Molecular modeling of energetic materials: the parameterization and validation of nitrate esters in the COMPASS force field. *J. Phys. Chem. B* **2000**, *104*, 2477−2489.
(74) Sun, H. COMPASS: an ab initio force-field optimized for condensed-phase applications overview with details on alkane and benzene compounds. *J. Phys. Chem. B* **1998**, *102*, 7338−7364.
(75) McQuaid, M. J.; Sun, H.; Rigby, D. Development and validation of COMPASS force field parameters for molecules with aliphatic azide chains. *J. Comput. Chem.* **2004**, *25*, 61−71.
(76) Kondratyuk, N. D.; Pisarev, V. V. Calculation of viscosities of branched alkanes from 0.1 to 1000 MPa by molecular dynamics methods using COMPASS force field. *Fluid Phase Equilib.* **2019**, *498*, 151−159.
(77) Rigby, D.; Roe, R. J. Molecular dynamics simulation of polymer liquid and glass. I. Glass transition. *J. Chem. Phys.* **1987**, *87*, 7285−7292.
(78) Yu, K. q.; Li, Z. s.; Sun, J. Polymer structures and glass transition: A molecular dynamics simulation study. *Macromol. Theory Simul.* **2001**, *10*, 624−633.
(79) Hadipeykani, M.; Aghadavoudi, F.; Toghraie, D. A molecular dynamics simulation of the glass transition temperature and volumetric thermal expansion coefficient of thermoset polymer based epoxy nanocomposite reinforced by CNT: A statistical study. *Phys. A (Amsterdam, Neth.)* **2020**, *546*, 123995.
(80) Buchholz, J.; Paul, W.; Varnik, F.; Binder, K. Cooling rate dependence of the glass transition temperature of polymer melts: Molecular dynamics study. *J. Chem. Phys.* **2002**, *117*, 7364−7372.
(81) Li, C.; Medvedev, G. A.; Lee, E.-W.; Kim, J.; Caruthers, J. M.; Strachan, A. Molecular dynamics simulations and experimental studies of the thermomechanical response of an epoxy thermoset polymer. *Polymer* **2012**, *53*, 4222−4230.
(82) Mohammadi, M.; Davoodi, J.; et al. The glass transition temperature of PMMA: A molecular dynamics study and comparison of various determination methods. *Eur. Polym. J.* **2017**, *91*, 121−133.
(83) Afzal, M. A. F.; Browning, A. R.; Goldberg, A.; Halls, M. D.; Gavartin, J. L.; Morisato, T.; Hughes, T. F.; Giesen, D. J.; Goose, J. E. High-Throughput Molecular Dynamics Simulations and Validation of Thermophysical Properties of Polymers for Various Applications. *ACS Appl. Polym. Mater.* **2021**, *3*, 620−630.
(84) Alcobaca, E.; Mastelini, S. M.; Botari, T.; Pimentel, B. A.; Cassar, D. R.; de Carvalho, A. C. P. d. L. F.; Zanotto, E. D. Explainable machine learning algorithms for predicting glass transition temperatures. *Acta Mater.* **2020**, *188*, 92−100.
(85) Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; Li, Y. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, PMLR: 2020; pp 1725−1735.
(86) Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. *Interspeech 2010* **2010**, 1045.
(87) Yi, Z. Evaluation and Implementation of Convolutional Neural Networks in Image Recognition. *J. Phys.: Conf. Ser.* **2018**, *1087*, 062018.
(88) Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, PMLR: 2018; pp 2323−2332.
(89) Kwon, Y.; Lee, D.; Choi, Y.-S.; Shin, K.; Kang, S. Compressed graph representation for scalable molecular graph generation. *J. Cheminf.* **2020**, *12*, 1−8.
(90) Jin, W.; Barzilay, R.; Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs In *International Conference on Machine Learning*, PMLR: 2020; pp 4839−4848.
(91) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A., Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *arXiv preprint* 2017; arXiv:1705.10843.
(92) Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; Aspuru-Guzik, A., Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *ChemRxiv* 2017, *2017*.
(93) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **2018**, *4*, 268−276.

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413

(94) Yan, C.; Wang, S.; Yang, J.; Xu, T.; Huang, J. Re-balancing variational autoencoder loss for molecule sequence generation. *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* **2020**, 1−7.
(95) Liu, K.; Sun, X.; Jia, L.; Ma, J.; Xing, H.; Wu, J.; Gao, H.; Sun, Y.; Boulnois, F.; Fan, J. Chemi-Net: a molecular graph convolutional network for accurate drug property prediction. *Int. J. Mol. Sci.* **2019**, *20*, 3389.
(96) Wang, X.; Li, Z.; Jiang, M.; Wang, S.; Zhang, S.; Wei, Z. Molecule property prediction based on spatial graph embedding. *J. Chem. Inf. Model.* **2019**, *59*, 3817−3828.
(97) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional embedding of attributed molecular graphs for physical property prediction. *J. Chem. Inf. Model.* **2017**, *57*, 1757−1772.
(98) Miccio, L. A.; Schwartz, G. A. From chemical structure to quantitative polymer properties prediction through convolutional neural networks. *Polymer* **2020**, *193*, 122341.
(99) Chen, G.; Tao, L.; Li, Y. Predicting Polymer's Glass Transition Temperature by A Chemical Language Processing Model. *Polymers* **2021**, *13*, 1898.
(100) Weyland, H.G.; Hoftyzer, P.J.; Van Krevelen, D.W. Prediction of the glass transition temperature of polymers. *Polymer* **1970**, *11*, 79−87.
(101) Yang, Y.; Zou, X.; Ye, H.; Zhu, W.; Dong, H.; Bi, M. Modified Group Contribution Scheme to Predict the Glass-Transition Temperature of Homopolymers through a Limiting Property Dataset. *ACS omega* **2020**, *5*, 29538−29546.
(102) Camacho-Zuniga, C.; Ruiz-Trevino, F. A new group contribution scheme to estimate the glass transition temperature for polymers and diluents. *Ind. Eng. Chem. Res.* **2003**, *42*, 1530−1534.
(103) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513−530.
(104) Gubaev, K.; Podryabinkin, E. V.; Shapeev, A. V. Machine learning of molecular properties: Locality and active learning. *J. Chem. Phys.* **2018**, *148*, 241727.
(105) Dunn, A.; Wang, Q.; Ganose, A.; Dopp, D.; Jain, A. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Comput. Mater.* **2020**, *6*, 1−10.
(106) Pirotti, F.; Sunar, F.; Piragnolo, M. BENCHMARK OF MACHINE LEARNING METHODS FOR CLASSIFICATION OF A SENTINEL-2 IMAGE. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2016**, *XLI-B7*, 335.
(107) Khan, J. Y.; Khondaker, M.; Islam, T.; Iqbal, A.; Afroz, S., A benchmark study on machine learning methods for fake news detection. *arXiv preprint* 2019; arXiv:1905.04749.
(108) Haque, F. M.; Grayson, S. M. The synthesis, properties and potential applications of cyclic polymers. *Nat. Chem.* **2020**, *12*, 433−444.
(109) Fox, T.; Loshaek, S. Influence of molecular weight and degree of crosslinking on the specific volume and glass temperature of polymers. *J. Polym. Sci.* **1955**, *15*, 371−390.
(110) Yamada, H.; Liu, C.; Wu, S.; Koyama, Y.; Ju, S.; Shiomi, J.; Morikawa, J.; Yoshida, R. Predicting materials properties with little data using shotgun transfer learning. *ACS Cent. Sci.* **2019**, *5*, 1717−1730.
(111) Blum, L. C.; Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **2009**, *131*, 8732−8733.
(112) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **2012**, *108*, 058301.
(113) Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864−2875.
(114) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 1−7.
(115) Qian, Z.; Cao, Z.; Galuska, L.; Zhang, S.; Xu, J.; Gu, X. Glass transition phenomenon for conjugated polymers. *Macromol. Chem. Phys.* **2019**, *220*, 1900062.
(116) Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P., Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint*; 2015 arXiv:1509.09292.
(117) Lusci, A.; Pollastri, G.; Baldi, P. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *J. Chem. Inf. Model.* **2013**, *53*, 1563−1575.

https://doi.org/10.1021/acs.jcim.1c01031
J. Chem. Inf. Model. 2021, 61, 5395−5413