JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

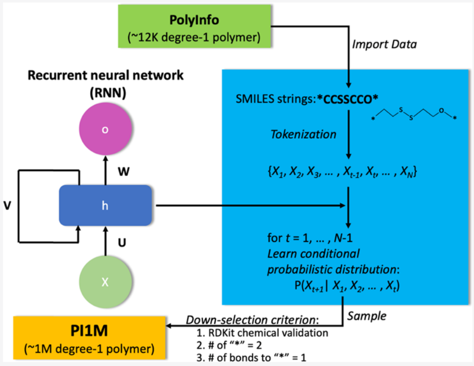
PI1M: A Benchmark Database for Polymer Informatics

Ruimin Ma and Tengfei Luo*

Cite This: *J. Chem. Inf. Model.* 2020, 60, 4684−4690 Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information

ABSTRACT: Open-source data on large scale are the cornerstones for data-driven research, but they are not readily available for polymers. In this work, we build a benchmark database, called PI1M (referring to ∼1 million polymers for polymer informatics), to provide data resources that can be used for machine learning research in polymer informatics. A generative model is trained on ∼12 000 polymers manually collected from the largest existing polymer database PolyInfo, and then the model is used to generate ∼1 million polymers. A new representation for polymers, polymer embedding (PE), is introduced, which is then used to perform several polymer informatics regression tasks for density, glass transition temperature, melting temperature, and dielectric constants. By comparing the PE trained by the PolyInfo data and that by the PI1M data, we conclude that the PI1M database covers similar chemical space as PolyInfo, but significantly populate regions where PolyInfo data are sparse. We believe that PI1M will serve as a good benchmark database for future research in polymer informatics.



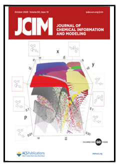
INTRODUCTION

Material informatics uses machine learning approaches to fast screen material candidates or generate new materials meeting certain criteria, so as to reduce the time of material development.[1] When the materials of interest are polymers, such data-driven machine learning studies can be called polymer informatics.[2] While machine learning algorithms[3,4] and platforms (e.g., scikit-learn,[5] PyTorch[6]) used for other fields in materials informatics should be equally applicable to polymer informatics, a grant challenge for polymer informatics is the lack of easily accessible databases. A number of polymer databases and/or platforms, like PolyInfo,[7] Polymer Genome,[8] CHEMnetBASE-Polymers,[9] have been developed, and several studies of polymer informatics based on these databases have been conducted.[8,10,11] However, most of these polymer databases are embedded in web applications, and the raw data (especially the data of chemical structures) are not accessible in large scale, thus conducting machine learning research is largely limited to the raw-data holders, which potentially impose barriers to test or develop machine learning algorithms for polymer informatics.

Large-scale open-source databases that are easily accessible in raw data form can lead to the prosperity of a branch of materials informatics, with one of the mostly studied open-source database ZINC[12] as a good example for pharmaceutical, biological, and small molecular research. For example, in 2018, leveraging the ZINC database, a variational autoencoder-based molecular generator was developed to automatically produce novel molecules from the latent chemical space, where the first gradient-based molecular design algorithm was established.[13] Later on, also using the ZINC database, constraint Bayesian optimization-based variational autoencoder[14] was proposed to improve the validity of generated molecules to address the problem that molecules selected by Bayesian optimization can lie in the "dead regions" of the latent space far away from the data used for training, as was mentioned in ref 13. Besides autoencoder, reinforcement learning was introduced, also based on studying the ZINC database, as an alternative approach for molecular design, using which multiple optimal molecules can be potentially found.[15] The availability of large-scale open-source databases can also greatly facilitate studies of representations, which is critical to material informatics. For example, in 2018, a machine learning representation of small organic molecules, molecular embedding, was developed based on learning the data in the ZINC database,[16] where the estimation of substructure similarity was properly implemented, which was not possible in traditional representations, like Morgan fingerprint.[17] The performance of quantitative structure-property-relationships can be improved using molecular embedding compared to those using Morgan fingerprint.

There are a large number of other machine learning research activities based on the ZINC database, as reflected by its citation count of 1757 since its first publication in 2012 (retrieved from Google Scholar on June 8, 2020). In contrast, the largest web-based polymer database, PolyInfo,[7] published

Received: June 25, 2020
Published: September 28, 2020



https://dx.doi.org/10.1021/acs.jcim.0c00726

in 2011, has only been cited for 18 times (retrieved from Google Scholar on June 8, 2020), where the abovementioned algorithms or studies have been rarely reported in polymer informatics, most likely due to the difficulties in accessing the data. This largely impedes the development of research in polymer informatics and its applications.

Thus, we are motivated to introduce a benchmark database in this work to help promote machine learning research in polymer informatics. We manually collect around 12 thousand (12k) polymer structures from the largest polymer database, PolyInfo, and then use them to train a generative model to generate around 1 million (1M) polymer structures outside of the training data set, and we name this new database PI1M (i.e., 1M polymers for polymer informatics). A machine learning representation for polymers, polymer embedding (PE), is introduced, which is then used to perform several polymer informatics regression tasks for density, glass transition temperature, melting temperature, and dielectric constants. By comparing the PE trained by the PolyInfo data and that by the PI1M data, we conclude that the PI1M database covers similar chemical space as PolyInfo, but significantly populate regions where PolyInfo data are sparse. The difference between the PI1M and the ZINC databases, as well as the difference between polymer and small organic molecules, has also been analyzed. We hope PI1M will serve as a benchmark testbed for future data-driven research in polymer informatics.

■ METHODS

Building PI1M. The process of building the PI1M database is shown in Figure 1. A generative model was used to learn the data distribution of PolyInfo, from which the structure files of ~12k degree-1 polymers (i.e., monomers) were collected manually, and then ~1M degree-1 polymers were sampled from this generative model to create the PI1M database. The details are further discussed below.

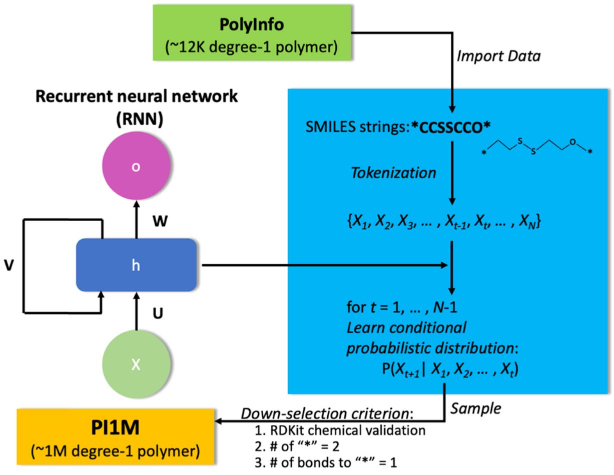


Figure 1. Process of building the PI1M database.

The degree-1 polymers in the PolyInfo are then represented in the format of SMILES strings,[18] e.g., "*CCSSCCO*", as shown in Figure 1, which is done using RDKit[19] to convert the structure file into its corresponding canonical SMILES strings. The "*" symbol was originally contained in the structure file of PolyInfo. Unlike common SMILES strings for small molecules, these polymer-SMILES strings contain distinct symbols of "*" to indicate the polymerization points of monomers, even though "*" can also be used for other purposes, like a wildcard atom. As a result, for the PI1M database, all "*" symbols in the SMILES correspond to the polymerization points. We call those "*"-decorated SMILES polymer-SMILES (p-SMILES), which are sequential data in nature. Only the p-SMILES with two "*" are used for training here, as they are the majority (99.953%) of the collected PolyInfo data. The generative model used is a recurrent neural network (RNN),[20] which is an architecture of neural networks designed to predict the future output given the present input and the memory about the past. As shown in Figure 1, for any given step, the o-cell (future output) is a result of the h-cell (hidden state) at the previous step (i.e., memory about the past) and the current input x-cell (present input), where W, U, and V are parameters. Such a feature makes the RNN particularly well suited for learning sequential data, where the future information is highly conditioned on the past memory and current input. In the case of p-SMILES strings, the unique sequences are the direct results of atomic connectivity, which makes it capable of capturing the fundamental chemistry of polymers. The RNN employed in our work was composed of three layers with 512 gated recurrent units in each layer, the dimension of word embedding was 128, and the sequence length used in RNN was decided by the maximum sequence length of the batch data used for training. The model was trained using gradient descent with a batch size of 128 and Adam[21] optimizer with an initial learning rate of 0.001 and a 0.03 decay in learning rate every 30 steps. PyTorch[6] was used to implement the model.

The p-SMILES strings are tokenized into a sequence of characters (X_1, X_2, X_3, ..., X_{t-1}, X_t, ..., X_N), where two additional tokens of "GO" and "EOS" are added for the RNN to initialize and terminate a polymer generation. For example, "*CCSSCCO*" is tokenized into ("GO", "*", "C", "C", "S", "S", "C", "C", "O", "*", "EOS"). These sequences are then encoded as sequences of one-hot vectors, which are used as inputs for RNN. At each step, RNN is used to learn the conditional probabilistic distribution, $P(X_{t+1}|X_1, X_2, ..., X_t)$, where X_t is the current input and (X_1, X_2, ..., X_{t-1}) are implicitly encoded in the hidden state h_{t-1}. The aim of the training is to maximize the likelihood assigned to the correct token, and the network parameters are updated by back-propagation through time.[22] Once the RNN has been trained on the p-SMILES, it can be used to generate new p-SMILES that follow the conditional probability distribution mentioned above. Specifically, the first input token ("GO") is given to initialize a sequence, and at each timestep, we sample an output token from $P(X_{t+1}|X_1, X_2, ..., X_t)$, and then the output token is used as our next input. Once the "EOS" token is sampled, the p-SMILES string is considered finished.

The training of RNN is conducted iteratively in three rounds, where ~12k degree-1 polymers from PolyInfo are used for training RNN in the first round, and then different numbers of generated degree-1 polymers are added for training in the following two rounds. Since the generated polymers are not guaranteed to be valid from the chemical perspective, we implement certain criteria to improve the validity of the polymers. We first validate whether the p-SMILES strings are chemically valid using RDKit, which reads the generated p-SMILES and can convert them into molecular objects if they are chemically valid. Then, we count the number of polymerization points ("*") in those p-SMILES that passed the verification by RDKit, and we remove them with the

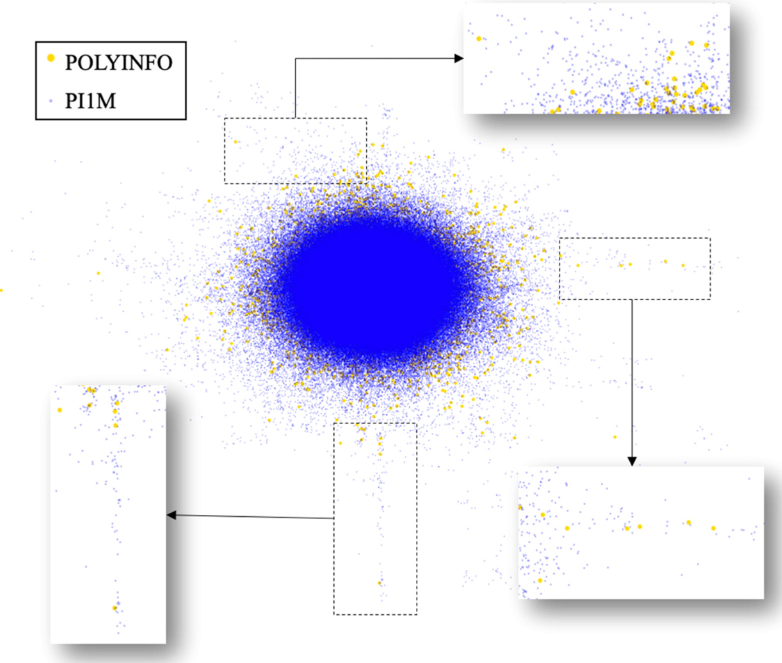


Figure 2. t-SNE visualization of m-PE of all polymers in PolyInfo and PI1M. The three insets highlight how PI1M data are filling up the chemical spaces between PolyInfo data.

number of "*" not equaling to 2 (note: all p-SMILES in training data have two "*"). Lastly, we check the number of bonds connected to "*" and keep the p-SMILES if the number of bonds connected to "*" is 1 (note: number of bonds connected to "*" is 1 for all p-SMILES in training data). The last two criteria imposed are based on the nature of the training data, which are for linear polymers. All generated polymers used for training RNN in the second and third rounds are those that meet the above criteria. The performance of RNN in each round is evaluated by the valid ratio of p-SMILES generated. Based on the results shown in Figure S1 of the Supporting Information, the valid ratio increases as more generated p-SMILES are added for training, which demonstrates that the amount of effective information increases as training data size increases. We generate ∼1M valid degree-1 polymers in total after the removal of duplicates and put them into our database PI1M, i.e., PI1M database consists of ∼1M p-SMILES, and those p-SMILES are all in their canonical format after going through RDKit. We note that none of degree-1 polymers collected in PI1M overlap with those in PolyInfo. Several degree-1 polymers from both PolyInfo and PI1M are shown in Figure S2 of the Supporting Information.

Polymer Embedding (PE). A numerical representation of polymers is an essential part of polymer informatics, since machine learning models always take digits as inputs when quantifying structure−property relationships. In terms of representing organic molecules, the most commonly used representations are Morgan fingerprints[17] and molecular embedding,[16] which have also been both used as polymer representations to quantify the relationship between polymer structures and properties.[23] Molecular embedding is proved to be better than Morgan fingerprints as a polymer representation, due to the accurate estimation of substructure similarity in molecular embedding.[23] Since monomers differ from small molecules in that they need to contain information of polymerization points to correctly represent the polymer chemistry, when molecular embedding was employed in ref 23, it was trained on two-monomer structures drawn manually, where the bonding information between neighboring monomers were explicitly included. In the PI1M database, all monomers contain the symbols "*" to include the polymerization points and thus the bonding information between monomers. The difference between two-monomer structures and "*"-decorated monomer structures can be found in Figure S3 of the Supporting Information. As a result, we call the molecular embedding trained on the polymer embedding (PE) of these monomers to distinguish from that for small molecules.

We obtain the PE by training the mol2vec model[16] on degree-1 polymers. In mol2vec, degree-1 polymers were first decomposed into a sequence of substructures, where the center substructure was used as input of a neural network to predict its surrounding substructures. Each substructure in degree-1 polymers was used as a center substructure for once and the polymer embedding was derived from the neural network's weights after training was done. For more technical details, please check ref 16. The embedding trained on PolyInfo data is denoted as POLYINFO-PE, while that trained on PI1M data is denoted as PI1M-PE; the embedding trained on pooled PolyInfo and PI1M data is called m-PE (i.e., mixed PE), and they are all 300-dimensional vectors, one example of which can be found in Figure S3 of the Supporting Information. In Figure 2, we visualize the m-PE of monomers in both PolyInfo and PI1M in the two-dimensional space using t-SNE.[24] From the plot, the chemical space of PolyInfo is well covered by the degree-1 polymers from PI1M, and the PI1M's degree-1 polymers are filling up the space between PolyInfo's degree-1 polymers (i.e., blue dots filling up the space between golden dots, insets in Figure 2). These observations indicate good sampling from RNN and show that PI1M can serve as a similar or even better benchmark database in the absence of the PolyInfo data which are not easily accessible, as from the data-driven perspective, the uncertainty in polymer's chemical space is reduced by those newly generated population from PI1M.

Polymer Informatics Tasks. To evaluate the quality of the PE, we use it as the polymer representation to quantify the structure−property relationships in several existing data sets, where the polymer properties in different data sets are diverse. We first collect the data from ref 23, where the relationship between polymer structures and their corresponding density, melting temperature, and glass transition temperature have been quantified. These bulk polymer properties are measured from experiments. We use the newly established PE to in place of the molecular embedding in ref 23, while keeping the rest of the method unchanged, such as the machine learning algorithm (random forest[25]), model training scheme (fivefold cross-validation), evaluation metrics (coefficient of determination (R^2), mean squared error (MSE), mean absolute error (MAE)), etc. The results are summarized in Table 1, where the shaded results are from ref 23, while the rest are from the present work. Both POLYINFO-PE and PI1M-PE are examined as polymer representation. For the three bulk properties studied, both PEs yield prediction accuracies within the error bars of those from the molecular embedding trained using the two-monomer structures in ref 23. The performances of POLYINFO-PE and PI1M-PE in these structure−property relationships also show no statistically significant difference, suggesting that PI1M can serve as a benchmark database in place of PolyInfo for polymer informatics studies.

Table 1. Quantitative Structure−Property Relationships between Different Polymer Representations and Different Polymer Properties[a]

		Density	
Metrics	Molecular Embedding	POLYINFO-PE	PI1M-PE
R^2	0.701±0.093	0.683±0.101	0.657±0.127
MSE	0.010±0.004	0.010±0.004	0.011±0.005
MAE	0.065±0.012	0.066±0.013	0.070±0.014
		Melting temperature	
Metrics	Molecular Embedding	POLYINF-PE	PI1M-PE
R^2	0.684±0.054	0.682±0.060	0.688±0.051
MSE	3268.919±468.430	3290.241±548.820	3245.107±531.456
MAE	40.464±2.652	40.907±4.168	40.251±3.553
		Glass transition temperature	
Metrics	Molecular Embedding	POLYINF-PE	PI1M-PE
R^2	0.865±0.030	0.862±0.035	0.860±0.031
MSE	1709.877±445.498	1740.902±473.594	1768.089±431.649
MAE	28.012±3.300	28.142±3.535	28.209±2.902

[a] The shaded results are from ref 23, which uses molecular embedding trained on two-monomer structures. The unshaded results are from the present study.

Another data set that we are able to collect is from ref 26, which contains 1073 polymers and their corresponding dielectric constants calculated from first-principles in the crystalline phase. The original data in this data set are in the crystallographic information format (CIF), and 701 data are able to be converted into SMILES strings. After removing outliers, 661 data are left, which are then used for the structure−property relationship study. The dielectric constant distribution of the 701 data and details about outlier removal can be found in Figure S4 of the Supporting Information. Again, both POLYINFO-PE and PI1M-PE are used as the polymer representations. Support vector machine (SVM)[27] is employed as the machine learning model (the best one among the machine learning algorithms being tested in Table S1 of the Supporting Information), where the radial basis function is used, the penalty parameter (C) of the error term is set to 20, and the value (ε) that specifies the penalty-free area is set to 0.2. The model is trained using fivefold cross-validation, and R^2, MSE, and MAE are used as the evaluation metrics.

The dielectric constant distribution of those 661 data is shown in Figure 3a, and the performances of the two different PEs are shown in Figure 3b, which also visualizes predictions versus ground truths on the validation sets. Reasonable performances with $R^2 > 0.5$ in both cases are achieved. All evaluation metrics from the PI1M-PE and POLYINFO-PE cases show no statistically significant differences from each other. Another two machine learning models, including random forest and multilayer perceptron,[28,29] have also been tested, and similar observations on their performances are found as seen in Table S1 of the Supporting Information.

Since ref 26 did not report any machine learning study based on its data set, we cannot perform a direct comparison. However, Kim et al.[8] used a portion of the data set in ref 26 and conducted machine learning studies, which serve as a reference for us to compare our machine learning performances with. In Kim's work,[8] 384 non-metal-containing

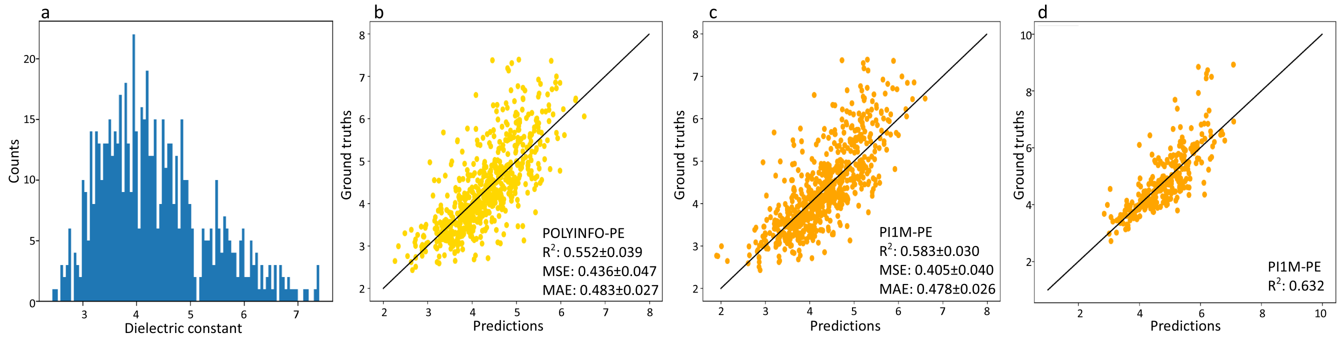


Figure 3. (a) Dielectric constant distribution of 661 polymers used for quantifying the structure−property relationship; (b, c) parity plots of model prediction and ground truth as well as the performances for POLYINFO-PE and PI1M-PE, respectively; (d) model performance using PI1M-PE and 306 non-metal-containing organic polymers for training.

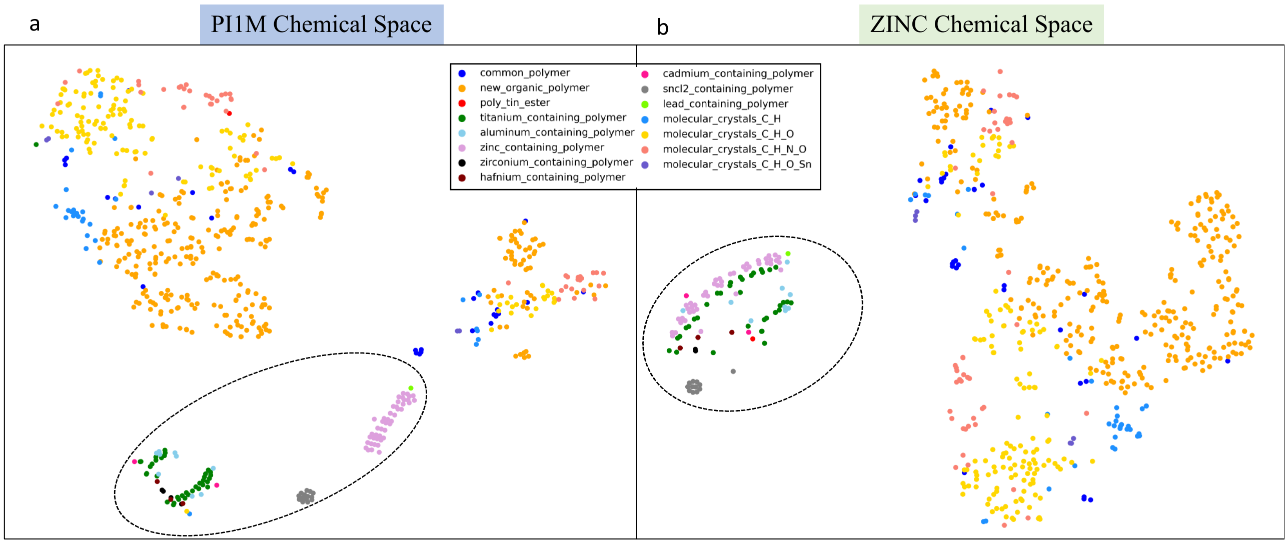


Figure 4. t-SNE visualization of 701 polymers represented by (a) PI1M-PE and (b) molecular embedding trained on ZINC molecules. Circled regions include X-containing polymers, where X = metal elements, or $SnCl_2$.

polymers with dielectric constants in the range of 2.61−9.09 were used to study the relationship between polymer structure and dielectric constant. The model was trained in using fivefold cross-validation, and R^2 was used as the performance evaluation metric. We can only find 314 non-metal-containing organic polymers in the data set from ref 26, and 306 of them have dielectric constants between 2.61 and 9.09, which are a subset of that used in Kim's work. We train an SVM ($C = 30$, $\varepsilon = 0.2$) using these 306 data using fivefold cross-validation and found that the average $R^2 = 0.632$ is comparatively lower than average $R^2 = 0.815$ from Kim's work. The difference is potentially due to the size of data set used in these two studies. Kim et al.[8] had more non-metal-containing polymers than the data set from ref 26. However, since the extra data used by Kim et al.[8] are not publicly available, we cannot confirm. Another potential reason is that the polymers studied in refs 8 and 26 are crystalline polymers; it is possible that the structure−property relationship can be better described using more explicit descriptors that are related to crystals, like the hierarchical fingerprint used in Kim's work. Nevertheless, using PI1M-PE, the SVM model performance of $R^2 = 0.632$ shows acceptable accuracy, although addressing the above problems might further improve the model.

Another piece of information from the data set of ref 26 is that polymers come from different polymer classes, and we list all of the 15 polymer classes that the total of 701 polymers belong to as defined in ref 26, including common polymers, new organic polymers, poly(tin ester), titanium (Ti)-containing polymers, aluminum (Al)-containing polymers, zinc (Zn)-containing polymers, zirconium (Zr)-containing polymers, hafnium (Hf)-containing polymer, cadmium (Cd)-containing polymer, $SnCl_2$-containing polymer, lead (Pb)-containing polymer, molecular crystals of C and H, molecular crystals of C, H, and O, molecular crystals of C, H, N, and O, and molecular crystals of C, H, O, and Sn. Polymers are separated into different classes due to their chemical structures or elements, and properly separating them in the chemical space may also be used as a criterion to evaluate the quality of a polymer representation. In addition, the ability to classify polymers based on their location in the chemical space can also help researchers focus on the study of a class of polymer of interest. We represent all 701 polymers with PI1M-PE, which are labeled with their corresponding polymer classes, and then visualize them in the two-dimensional space by t-SNE, as shown in Figure 4a. From the plot, we can see that polymers group into several obvious clusters in the chemical space. The common polymers, new organic polymers, and molecular crystals are well separated from the X-containing polymers (X = metal elements, or $SnCl_2$). In addition, several clusters are also clearly observed in the X-containing polymers due to that they contain different elements or compounds. To further gauge PI1M-PE as a polymer representation, we compare the results to those yielded by the widely used molecular embedding,[16] which is trained on 20 million ZINC[12] druglike small molecules (Figure 4b). The motivation of comparing PI1M and ZINC is: ZINC molecules and PI1M molecules are both organic molecules; they do share some common substructures and the logic of substructures integrating into organic molecules; the molecular embedding obtained from ZINC database and polymer embedding obtained from PI1M database via mol2vec are all substructure-driven; if the substructure similarity is similarly estimated in both databases, then the knowledge can be transferrable between small druglike molecules and polymers, and ref 23 has proved that molecular embedding trained on ZINC database can be used for predicting polymers' properties. The molecular embedding trained on ZINC molecules can also separate common polymers, new organic polymers, and molecular crystals from those X-containing polymers, but it cannot separate X-containing polymers with different elements as well as PI1M-PE. When analyzing these two databases, we find that the ZINC molecules used for training[16] only contain elements H, C, N, O, F, S, P, Cl, B, and Br; thus, metal elements are ignored when using the molecular embedding as a representation, which explains why those X-containing polymers are all in the close neighborhood of each other (dots in the black cycle in Figure 4b). However, PI1M polymers contain a wider range of elements, including Ca, K, P, Sn, N, S, Cd, Si, Li, Ni, As, Na, I, Ge, Fe, Co, Te, Zn, B, O, C, H, F, Cl, and Br, which can reasonably separate Zn-, $SnCl_2$- and Cd-containing polymers from each other, even though the other metal element-containing polymers cannot be well separated since those elements do not exist in PI1M. Thus, PI1M-PE may be more suitable as the polymer representation than the molecular embedding trained on druglike small molecules. Of course, there is also a need to add more diverse polymers into training the RNN to further expand the chemical space covered by PI1M, which will be our future study.

■ CONCLUSIONS AND DISCUSSION

In summary, we built a benchmark database PI1M, which may be used for machine learning studies in polymer informatics, as easily accessible open-source data in large scale plays a critical role promoting data-driven research in this field. We discussed in detail how PI1M was created using an RNN and compared its performance with PolyInfo, the database that PI1M is derived from, but the raw data of which are not easily accessible. A new machine learning representation, PE, was introduced, and several example polymer informatics tasks have been conducted to show the potential of PI1M-PE as a general-purpose polymer representation, since it was obtained without any bias from the labels. The potential benefit of PI1M for polymer informatics is further demonstrated via the comparison to ZINC, as the chemical elements in polymers can be much more diverse than that in druglike small molecules. Based on all of the abovementioned studies, we believe that PI1M can serve as an important benchmark database for future research in polymer informatics. PI1M is built for informatics purpose here, but it has some openings for other research purposes as well. We are aware of the rare cases (5 over ~1M); here, that degree-1 polymers "*CC*" and "*CCCC*" are coexisted in PI1M, which will become the same polymer chain after the polymerization. There might be some other chemical concerns besides this, as the number of PolyInfo polymers (~12k) used for training is still limited when in comparison to the number of possible polymers that could exist in nature. In other words, the chemical rules of encoding polymers covered by the training data here are not capable of encoding all of the possible polymers, but it is proven to be a good alternative to PolyInfo. PI1M will be published without polymer properties, but the polymers in it can be labeled by researchers depending on their need, either computationally or experimentally, which can take many years to accumulate.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c00726.

> Valid ratio of p-SMILES generated as a function of training epochs in different rounds (Figure S1); visualization of selected degree-1 polymers (monomers) from both PolyInfo and PI1M (Figure S2); (a) "*"-containing monomer structure used in this study, (b) two-monomer structure used in ref 23, and (c) example of polymer embedding (Figure S3); (a) dielectric constant distribution of the 701 data and (b) definition of outliers used in the study (Figure S4); quantitative relationship between polymer structures and their corresponding dielectric constants (Table S1) (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Tengfei Luo − *Department of Aerospace and Mechanical Engineering and Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, Indiana 46556, United States;* orcid.org/0000-0003-3940-8786; Email: tluo@nd.edu

Author

Ruimin Ma − *Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, Indiana 46556, United States;* orcid.org/0000-0003-1527-9289

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c00726

Notes

The authors declare no competing financial interest.
PI1M database will be available at the repo https://github.com/RUIMINMA1996/PI1M upon thepublishing of this work. The dataset can be used freely for academic purposes.

■ ACKNOWLEDGMENTS

The computation is supported in part by the University of Notre Dame, Center for Researching Computing, and NSF through XSEDE resources provided by TACC Stampede II under a grant number TG-CTS100078.

■ REFERENCES

(1) Hill, J.; Mulholland, G.; Persson, K.; Seshadri, R.; Wolverton, C.; Meredig, B. Materials science with large-scale data and informatics: unlocking new opportunities. *MRS Bull.* **2016**, *41*, 399−409.

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

(3) James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*; Springer: New York, 2013;Vol. *112*, pp 3−7.

(4) Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.; Kim, C. Machine learning in materials informatics: recent applications and prospects. *npj Comput. Mater.* **2017**, *3*, No. 54.

(5) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(6) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A. PyTorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026−8037.

(7) Otsuka, S.; Kuwajima, I.; Hosoya, J.; Xu, Y.; Yamazaki, M. In *PoLyInfo: Polymer Database for Polymeric Materials Design*, 2011 International Conference on Emerging Intelligent Data and Web Technologies, 2011; pp 22−29. https://polymer.nims.go.jp/en/.

(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) CHEMnetBASE-Polymers. A Property Database. 2017, http://poly.chemnetbase.com.

(10) Wu, S.; Kondo, Y.; Kakimoto, M. A.; Yang, B.; Yamada, H.; Kuwajima, I.; Lambard, G.; Hongo, K.; Xu, Y.; Shiomi, J.; Schick, C.; et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *npj Comput. Mater.* **2019**, *5*, No. 66.

(11) 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.

(12) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.

(13) 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.

(14) Griffiths, R. R.; Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.* **2020**, *11*, 577−586.

(15) Popova, M.; Isayev, O.; Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **2018**, No. eaap7885.

(16) Jaeger, S.; Fulle, S.; Turk, S. Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition. *J. Chem. Inf. Model.* **2018**, *58*, 27−35.

(17) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(18) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.

(19) Landrum, G. RDKit: Open-Source Cheminformatics Software *GitHub and SourceForge*, 2016, *10*, 3592822.

(20) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.

(21) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. 2014. arXiv:1412.6980. arXiv.org e-Print archive. https://arxiv.org/abs/1412.6980.

(22) Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proc. IEEE* **1990**, *78*, 1550−1560.

(23) 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.

(24) Maaten, L. V. D.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.

(25) Breiman, L. Random forests, machine learning. *J. Clin. Microbiol.* **2001**, *45*, 5−32.

(26) Huan, T. D.; Mannodi-Kanakkithodi, A.; Kim, C.; et al. A polymer dataset for accelerated property prediction and design. *Sci. Data* **2016**, *3*, No. 160012.

(27) Chang, C. C.; Lin, C. J. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1−27.

(28) Hinton, G. E. Connectionist Learning Procedures. In *Machine learning*; Morgan Kaufmann, 1990; pp 555−610.

(29) Vinutha, H. P.; Poornima, B.; Sagar, B. M. Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. In *Information and Decision Sciences*; Springer, 2018; pp 511−518.