Graph convolutional network for predicting properties of polymers

Carlos Matherson[1, a)]

Science Undergraduate Laboratory Internship, Environmental Science Division, Argonne National Laboratory, Lemont, Illinois 60439, United States of America

(*Electronic mail: camather@asu.edu.)

(Dated: 5 August 2022)

Graph representations of molecules have been used to accurately predict the properties of polymers, in some cases with more success than typical feature-list molecular representations.[1] The purpose of this project is to recreate the paper published by Park et al, Prediction and Interpretation of Polymer Properties Using the Graph Convolutional Network, in order to affirm that graph representations of molecules can be used to predict polymer properties. By means of recreating the Park study, the relationship between thermal and mechanical properties and structural characteristics of monomer units is explored through implementing a graph convolutional network (GCN) to model and predict the glass transition temperature (T_g), melting temperature (T_m), and density (ρ) for polymers curated from the PolyInfo database. This project focuses on building a comparable graph convolutional network machine learning model that can reproduce the accuracy of the Park model and extended to applications beyond the scope of the Park study. Additionally, improvements for Park's machine learning model are proposed in this project.

[a)]https://github.com/carlosmatherson/PolymerGCN.

I. INTRODUCTION

Machine learning is a powerful technique of leveraging data that is particularly useful for application in computational chemistry. Significantly, machine learning can be applied to chemistry as a tool for inversely designing polymers tailored to a desired performance, whether it be with regard to degradation rate, environmental fate, or other mechanical or thermal properties for industrial use. Thus, machine learning models are often created to render predictions of properties efficiently and accurately.

In order to obtain such predictions, chemical compounds and molecules need to be represented in a machine-readable format. Graphs are one way to represent compounds in this way. When representing compounds as graphs, convolutional neural networks, a classification of artificial neural networks typically used to analyze imagery, can be modified to accept graph objects as input for regression.

The body of literature surrounding the application of machine learning, particularly the graph convolutional network (GCN) as described above, to polymers is limited and often expresses that graph representations of molecules are inferior to the more often used feature-list representations such as extended-connectivity fingerprints. For this reason, there is reason to doubt the application of GCN models to polymers.

In a 2022 study by Park, it is shown that GCN models that use graph representations can predict polymer properties as accurately or better than models that implement typical feature-list representations of molecules.[1] Because GCNs have the capacity to automatically learn task-specific representations using graph convolutions and do not require traditional hand-crafted descriptors or fingerprints,[2] it is worthwhile to recreate the Park study to verify the potential of GCN models and attempt to reproduce the accuracy seen in the Park study.

II. METHODS

A. Data collection

Park manually collected data for 2687 different structures of organic polyamides, an important class of polymers for engineering that exhibit high thermal stability and mechanical strength, from the PolyInfo open-access database. This data necessary to recreate the Park study is unattainable from the authors or the published supplementary information.

To overcome the issue posed by the unavailability of the source data, we gathered publicly available data from a study titled Graph Rationalization with Environment-based Augmentations by Liu et al (2022), which is also sourced from PolyInfo.[3] However, the data set from the Liu paper is not comprised of a particular type of polymer like the data set of polyamides used in Park. It is important to note that this difference in data may impact the results of a faithful recreation of the Park model, and so this is taken into consideration when analyzing the model evaluation results. To visualize the difference between the two data sets, the distribution of each property is presented in Figure 1.

Overall, the datasets are similar, but noticeable differences are seen between the sets of glass transition temperature and density data points. The Park glass transition data is multi-modal while the data set used in this project has a relatively normal distribution. Additionally, the density data set in this study seems to contain more outliers. Without access to the raw Park data, the analysis of data set variance is limited to visual comparison.

To create a workable data set, the raw data was filtered, selecting only the 445 data points that each have all three properties of interest and the SMILES notation of the monomer unit of the compound. This data set is used for each model presented throughout this project.

B. Molecular representation

Using the Spektral graph deep learning Python library requires data sets to be transformed into a list of graph objects that each contain node features, an adjacency matrix, edge features, and labels.[4] To transform each compound in the data set into machine-readable information and subsequently Spektral graph objects, we employed RDKit, a Python library for cheminformatics, to extract 5 atom features from the SMILES of each compound. For each compound, the atom type, number of hydrogen atoms, implicit valence, degree, and aromaticity are one-hot encoded to form a node-features matrix that is used as input for the graph convolutional network.[5]

The second input for the models is an adjacency matrix representing the intramolecular bonds of each compound. RDKit was also employed to construct these adjacency matrices using the SMILES representation of each compounds monomer unit.

Other features can be extracted from SMILES, but only those used by Park are used in this project. Edge features were not utilized because a fundamental characteristic of a GCN is that it does not require edge features. Instead, it uses an adjacency matrix to represent edge information.

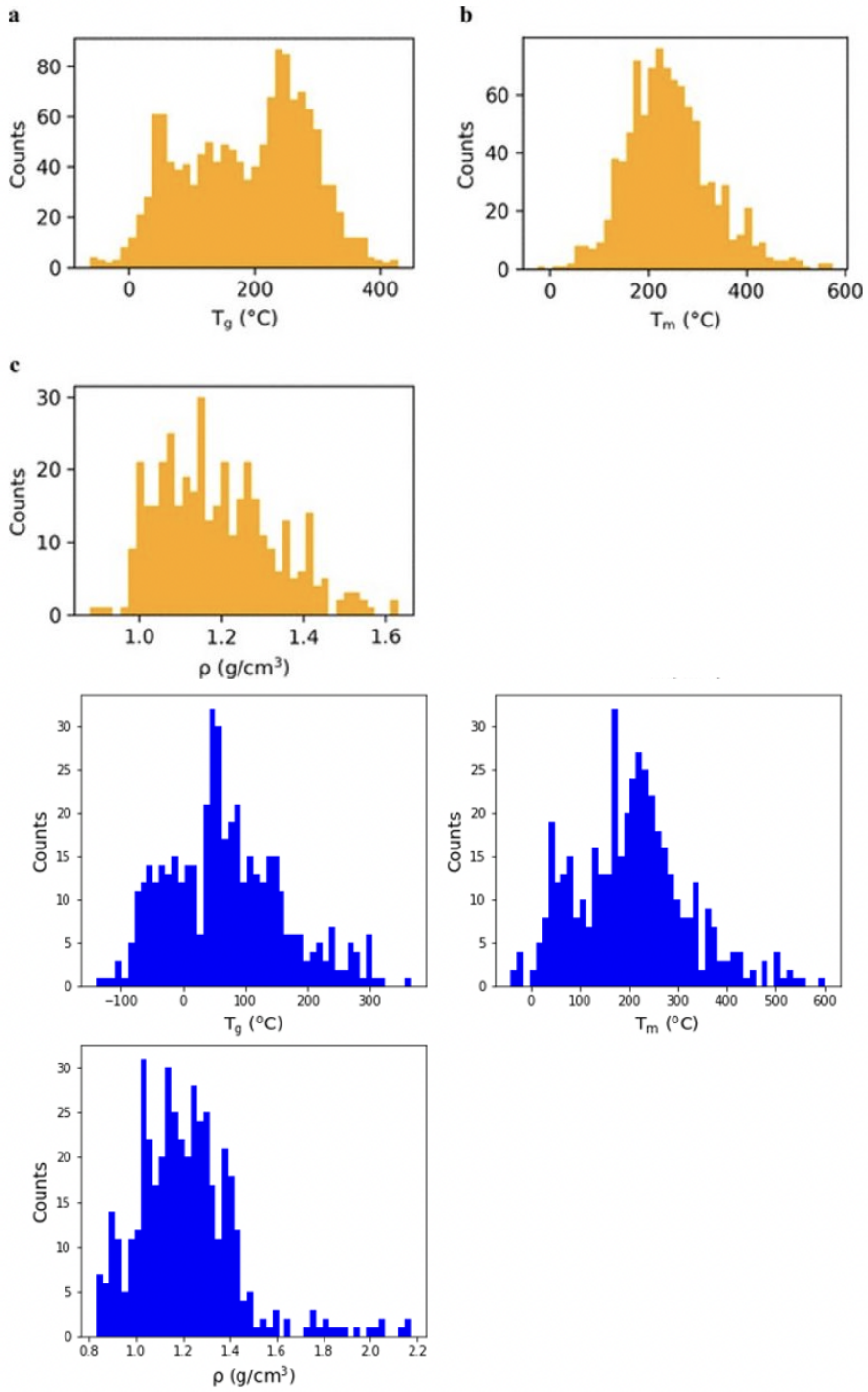

FIG. 1. In orange: data distributions of (a) glass transition temperature (T_g), (b) melting temperature (T_m), and (c) density (ρ) of 1388, 942, and 390 polyamides used in the Park study, respectively. In blue: data distributions of (top left) glass transition temperature (T_g), (top right) melting temperature (T_m), and (bottom) density (ρ) of 445 polymers curated from the Liu study for this project. Both data sets were collected from the PoLyInfo database.

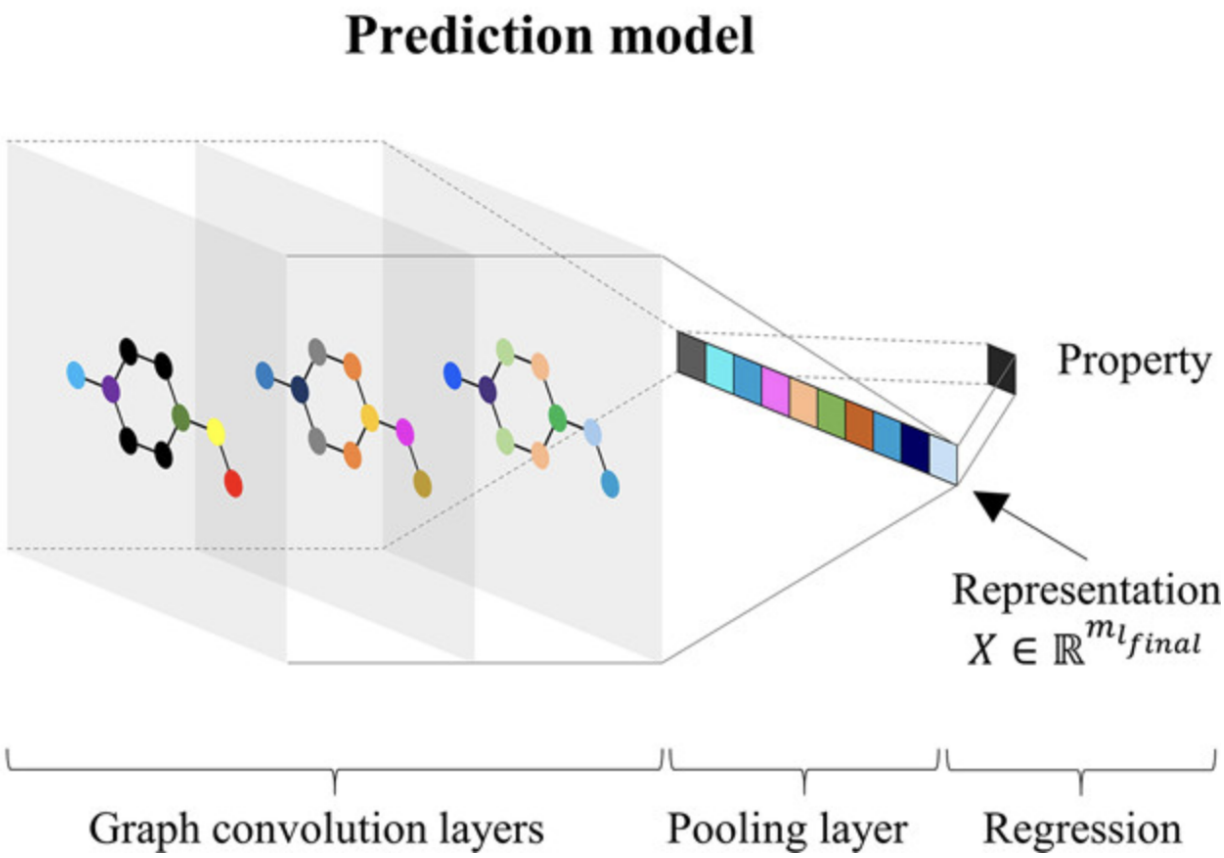

FIG. 2. Graph-level molecular representation vector X is obtained by the global sum pooling layer that sums up the feature vectors of all the atoms at the final convolution layer. The vector X is input to the regression algorithms (single dense layer or multilayer perceptrion) for the property prediction.[1]

Model		Featurization	Regression	T_g		T_m		ρ	
				RMSE	R^2	RMSE	R^2	RMSE	R^2
Matherson	Graph Convolutional Network	Atom Features and Adjacency Matrix	Single Dense	38.30 (9.04)	0.77 (0.13)	57.73 (21.1)	0.65 (0.26)	0.13 (0.015)	0.55 (0.17)
			Neural Network	**36.28 (7.90)**	**0.80 (0.11)**	**50.02 (22.4)**	**0.73 (0.25)**	**0.12 (0.010)**	**0.68 (0.07)**
Park			Single Dense	34.09 (3.07)	0.87 (0.03)	44.81 (2.25)	0.70 (0.05)	0.073 (0.018)	0.58 (0.22)
			Neural Network	**29.98 (2.16)**	**0.90 (0.02)**	**40.37 (2.94)**	**0.76 (0.05)**	**0.064 (0.013)**	**0.70 (0.17)**

FIG. 3. Prediction performances of the machine learning models for the glass transition temperature (T_g), melting temperature (T_m), and density (ρ) using graph molecular representation and different regression algorithms (single dense or multilayer perceptron). The values are mean and standard deviation (in the parenthesis) over the 5-fold cross validation splits. The bold case indicates the best performances for each property per model. The units of RMSE are K and g/cm^3 for T and ρ, respectively.

C. Model Hyperparameters

The architecture for the GCN models is determined by the hyperparameters outlined in Park. Each of the three models consist of the same classifications of layers: graph convolutional layers, in which patterns within the graph representations are learned, global sum pooling layers to summarize the learned patterns, and densely connected multilayer perceptron (MLP) regression layers to learn and predict the properties associated with each polymer compound. Figure 2 depicts a high-level visualization at the architecture of the GCN model.

The glass transition temperature, melting temperature, and density models each contain 6, 3, and 1 GCN layers with 100 kernels, respectively. In addition to these layers, every model consists of 1 global sum pooling layer and 2 MLP layers. The MLP layers for glass transition temperature and density models contain 300 kernels, and only 100 kernels are used in the melting temperature model. Every layer also utilized L2-norm kernel regulizers to reduce the weights between layers. The parameter for this is set to 1e-2. The training hyperparameters also differed between each model. The learning rate for the temperature models is 1e-3, but it is slower for the density model at 1e-4. Each model is trained and evalutated over 4000 epochs with 5-fold cross validation and batch size of 15.

For each model, 2 outputs are generated in order to examine the performance of the GCN layers with a single dense output independent from the MLP layers and the complete model including the MLP regression algorithm.

III. RESULTS AND DISCUSSION

We present the results by comparing the performances of the GCN models. Figure 3 shows the prediction accuracies of our GCN model and the Park model with the different regression algorithms. It is clear from the table that, for each model, the MLP (labeled "Neural Network") regression is an improvement from the single dense layer regression across all properties.

The Park model outperforms our model across every property as well. This is likely due to the difference in the data set used or possibly differences in model implementation. Both are valid reasons since neither the source code nor data is available from Park. The closest performance metric between the models occurs between the density models. Our density model average R^2 value lies between the R^2 values of the two regression algorithms of the Park model.

GCN for predicting properties of polymers

Real physical trends are present in the results as we can see that density is not strongly correlated with the structure of the monomer units of a compound in either model. Glass transition temperature shows the strongest correlation between thermal property and structure in both models. These results make sense as density often relies on the macro-structure of polymer compounds, the presence of co-polymers, or other factors not accounted for in the monomer unit SMILES.

IV. CONCLUSIONS

Over all, both models performed similarly when the obstacles faced in reconstructing the model are taken into consideration, and the findings of the Park paper are affirmed. Recreating the Park GCN model confirms that graph representations can be used to predict polymer properties with decent accuracy. The results also reaffirm the relationship between monomer unit structure and overall polymer properties.

Recreating the Park study has lent ideas for improvement on the model. In future work, our model can be improved through tuning, adjusting the architecture, and training the model on a larger data set for a more robust result. Improvements on the Park study would include a more rigorous tuning of the model as Park does not use optimization techniques to select hyperparameters aside form the number of GCN layers. Methodically selecting optimal values for the number of kernels in each layer, epochs, learning rate, batch size, and other hyperparameters would lead to a formidable improvement on the exisiting GCN machine learning model.

Moreover, there are ways to extend the model to new applications. Future efforts may focus on customizing the model so that it can be used to predict all three, and potentially more, properties from the same graph representations. This type of model would be more powerful and have more utility than the 3 individual models, which are relatively limited in scope with respect to real-world application like the aforementioned inverse design of polymers.

ACKNOWLEDGMENTS

I wish to acknowledge the support of Eugene Yan, Jeremy Feinstein, Margaret MacDonell, and the SULI Polymer & Earth Systems Modeling team.

REFERENCES

[1]J. Park, Y. Shim, F. Lee, A. Rammohan, S. Goyal, M. Shim, C. Jeong, and D. S. Kim, "Prediction and Interpretation of Polymer Properties Using the Graph Convolutional Network," ACS Polymers Au (2022), 10.1021/acspolymersau.1c00050, publisher: American Chemical Society.

[2]D. Jiang, Z. Wu, C.-Y. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu, and T. Hou, "Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models," Journal of Cheminformatics **13**, 12 (2021).

[3]G. Liu, T. Zhao, J. Xu, T. Luo, and M. Jiang, "Graph Rationalization with Environment-based Augmentations," (2022), arXiv:2206.02886 [cs, stat].

[4]D. Grattarola and C. Alippi, "Graph Neural Networks in TensorFlow and Keras with Spektral," (2020), arXiv:2006.12138 [cs, stat].

[5]J. Park, Y. Shim, F. Lee, A. Rammohan, S. Goyal, M. Shim, C. Jeong, and D. S. Kim, en"Supporting Information: Prediction and Interpretation of Polymer Properties Using the Graph Convolutional Network," , 9.

[6]M. Dablander, en-US"How to turn a SMILES string into a molecular graph for Pytorch Geometric | Oxford Protein Informatics Group,".