




ARTICLE OPEN



Polymer graph neural networks for multitask property learning

Owen Queen^{1,2}, Gavin A. McCarver³, Saitheeraj Thatigotla^{1,2}, Brendan P. Abolins⁴, Cameron L. Brown⁴ , Vasileios Maroulas¹  and Konstantinos D. Vogiatzis³ 

The prediction of a variety of polymer properties from their monomer composition has been a challenge for material informatics, and their development can lead to a more effective exploration of the material space. In this work, POLYMERGNN, a multitask machine learning architecture that relies on polymeric features and graph neural networks has been developed towards this goal. POLYMERGNN provides accurate estimates for polymer properties based on a database of complex and heterogeneous polyesters (linear/branched, homopolymers/copolymers) with experimentally refined properties. In POLYMERGNN, each polyester is represented as a set of monomer units, which are introduced as molecular graphs. A virtual screening of a large, computationally generated database with materials of variable composition was performed, a task that demonstrates the applicability of the POLYMERGNN on future studies that target the exploration of the polymer space. Finally, a discussion on the explainability of the models is provided.

npj Computational Materials (2023)9:90; <https://doi.org/10.1038/s41524-023-01034-3>

INTRODUCTION

The ubiquitousness of polymers in modern technology highlights their importance in and for the modern world. The features that are responsible for their widespread use and applicability are favorable mechanical and thermal properties, high durability, and general resistance to corrosion¹. Within the broad category of polymeric materials, there is a subgroup of polymers known as polyesters, which are composed of ester repeat units^{2–4}. The most common polyester produced is polyethylene terephthalate (PET), which can be found in a host of applications including packaging, textiles, thermoplastic resins, and photovoltaic devices^{5–7}. PET is comprised of repeating units of terephthalic acid and ethylene glycol. The versatility of the esterification reaction allows many different types of multifunctional acids and glycols to be polymerized into polyesters and indeed some applications make use of multiple acids and multiple glycols in the same composition. Adding to this complexity are the various molecular weights and end-group distributions, as well as branching of the polymer backbone that have been realized. Together, this gives rise to an ever-increasingly large materials design space.

For the exploration of such a large materials space, machine learning (ML) can be utilized in order to derive highly nonlinear relationships between the polymeric materials and their corresponding properties^{8–16}. The glass transition temperature (T_g), molecular weight (MW), the molecular weight distribution or polydispersity index (PDI) and the inherent viscosity (IV) are such properties that are correlated with the functionality and performance of the material. In order to develop polyesters, which have favorable T_g , MW, and IV values for a given application, a costly and time-consuming approach is often utilized, which involves testing many different combinations of diacids and diols in different experimental setups, including synthesis conditions, catalyst selection, and different monomer ratios. This can take hours or days for a single batch and is not able to cover a large number of targeted materials in a single instance. To circumvent such demanding processes, ML can be utilized to map the correlation between a given structural input (identities of the

diacids and diols used) and output (the desired properties such as T_g , MW, and IV) in order to help guide experimental work on the targeted synthesis of materials with enhanced properties.

Previous work has demonstrated great performance of ML models on the prediction of glass transition temperatures^{14,16–20}. Tao and coworkers tested a large array of ML models with varying structure and feature representations in order to provide T_g predictions¹⁴. Using a dataset of about 7000 homopolymers, they developed a ML model with good predictability and were able to provide estimates on ~5700 homopolymers with unknown experimental T_g values. A similar study examined polyacrylamides with quantum chemical descriptors in order to provide T_g predictions¹⁶. A Gaussian process regression model was developed from a small dataset (20 instances) to estimate T_g using thermal energies and the total electronic energies of the repeat units as input values.

A task that still remains elusive for ML applications on polymers is the prediction of multiple properties by a single model, which can lead to more effective material optimization¹⁵. Another important challenge is the accurate prediction of IV values. ML models that are based solely on monomer composition ignore important structural information such as the number of end-groups or the polymer chain length (often approximated by the molecular weight) and struggle to differentiate between values within a narrow range (between 0.2 and 0.4 dL/g). In this range, the relationship between T_g and IV can vary substantially²¹ and while T_g accuracy is fairly straightforward, predictions for IV are more challenging. Few studies have targeted IV values using ML^{9,11} with limited success and thus, alternative methodologies should be considered. Recent work has shown that graph neural networks (GNNs) provide increased predictability regarding thermal and mechanical properties of polymers^{22,23}, on other families of materials^{24,25}, as well as on molecular properties^{26–30}. As a result of the success of GNNs, graphs provide a promising direction for representing molecules. Neural networks are particularly well suited to combine molecular graphs into macromolecules in a similar manner to how representation methods such as BigSMILES^{31,32} represents polymers. Another

¹Department of Mathematics, University of Tennessee, Knoxville, TN 37996-1320, USA. ²Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996-2250, USA. ³Department of Chemistry, University of Tennessee, Knoxville, TN 37996-1600, USA. ⁴Eastman Chemical Company, Kingsport, TN 37660, USA. ✉email: cameronl.brown@eastman.com; vmaroula@utk.edu; kvogiatz@utk.edu

approach that has been recently explored covers ML models on feature-engineered polymer data that capture higher-order structural interactions between monomeric units^{14,33}.

In this work, we have developed a multitask ML architecture that aims to provide reliable values for polymer properties. To access these complex structure-function correlations, we have utilized a GNN-based model (POLYMERGNN), which has been tested on a dataset of experimentally measured properties of polyesters (T_g and IV values). We demonstrate the generality of this model in its ability to predict T_g and IV as single tasks as well as both T_g and IV in a multitask learning framework. POLYMERGNN outperforms other molecular embedding techniques in the tested prediction tasks, while it has the ability to work in low-data availability regimes. In addition, we demonstrate the robustness of POLYMERGNN through an explainability study, showing that the model appears to learn chemically relevant patterns and features in the dataset. This proposed methodology, while demonstrated for polyester prediction, is transferable to other families of materials. While GNN-based models for machine learning predictions of polymer properties have been previously developed and successfully tested^{22,23,33}, the pooling mechanism introduced here further advances these models (*vide infra*). This mechanism creates a centralized vector enriched with information from all monomers and allows POLYMERGNN to make predictions on monomer input without any direct modeling of polymers.

RESULTS

The polymer database

A diverse database of polyester resins that includes experimental data was generated. These materials contain between 1 and 4 different diacids (referred to in the next paragraphs as acids) and between 1 and 4 diols (glycols), while a small number also includes trimethylolpropane (TMP) that allows the synthesis of branched polymers (Fig. 1a, the full list of all monomers is given at the Supplementary Note 1). The overall database contains 186 linear polymers (62.8%) and 110 branched polymers (37.2%). The linear polymers can be further classified as “homopolyesters”, which only have 1 acid and 1 glycol (24.0%), and “co-polyesters”, which have multiple acids and/or glycols (38.8%). A small percentage of the linear polyesters (21.3% of the total database) include high molecular weight polymers with a characterized amount of cyclic oligomers (referred in the next sections as “cyclic”). Pictorial representations of the subsets of polyester resins are shown in Fig. 1b–d. The polymer properties collected for each material include T_g , IV, and weight-average molecular weight (M_w) (as a function of polystyrene), acid number (AN) and hydroxyl number (OHN). Figure 1e, f, and g show the distribution of the T_g and IV values for each subset (linear, branched, and “cyclic”, respectively), while representative examples of the three subsets are given in Fig. 1h, i, and j, respectively. It is thus evident that the compiled database has extensive diversity with respect to the material composition and structure, as well as with respect to the targeted properties. In addition, not all data entries have measured T_g and IV values. 210 instances in the database contain measured T_g values, 243 instances have measured IV values, and 163 instances have both T_g and IV values.

Initial model analysis

Using the diverse polyester resin dataset, we initially performed a wide-scale study to examine how different machine learning architectures, molecular representations, and polyester chain lengths affected the prediction of T_g and IV values (Supplementary Note 2). With regards to the machine learning architecture, we found that the kernel ridge regression (KRR) method resulted in the highest or near-highest predicted R^2 values for T_g and IV with values of 0.8624 and 0.7067, respectively. From this study, we

found that the inclusion of M_w values in the input vector improves the ability to predict IV values significantly whereas it does not improve the prediction of T_g values: IV was predicted with R^2 values of 0.4288 and 0.7067 without and with M_w while T_g was predicted with R^2 values of 0.8624 and 0.8582 without and with M_w using the KRR model. We also found no systematic increase in property accuracy when lengthier oligomers were used as input to the ML model since the use of individual acids and glycols monomers resulted in the highest R^2 values for both T_g and IV. Thus, these are the values that will serve as a baseline for the POLYMERGNN architecture.

POLYMERGNN architecture

We introduce POLYMERGNN, a neural network and general training procedure to predict properties of polymers of known monomer composition. The overall data modality introduces a challenge as it is not straightforward to represent the polymer composition as simple vectors or mathematical objects that can naturally be input to machine learning algorithms. For that reason, POLYMERGNN leverages graph neural networks (GNNs) and a pooling mechanism to produce outputs for varying numbers of inputs (number of acids and glycols in a given resin). Importantly, POLYMERGNN separates acid and glycol inputs and combines representations from both of these sets of monomers to produce downstream representations with rich chemical information. As POLYMERGNN utilizes a neural network, it can also perform multitask learning to produce embedding vectors that are optimized for predicting multiple properties.

The full POLYMERGNN architecture consists of three separate units: (1) a molecular embedding block, (2) a central embedding block, and (3) a prediction network, as seen in Fig. 2. Each unit is presented separately in the following paragraphs.

Molecular embedding block

The molecular embedding block is responsible for transforming input molecular graphs into vectors, or representations of the molecular structures. Each resin is represented by its constituent monomers—initial inputs into the synthesis of the resin. The molecular structure of each monomer is then encoded into a molecular graph where nodes, or vertices, correspond to atoms, and edges correspond to chemical bonds. A GNN with two graph convolutional layers is used for each acid and glycol. Through rigorous testing of various GNN layers, we found that a two-layer GNN, with a Graph Attention Network (GAT) layer³⁴ followed by a GraphSAGE layer³⁵, provided exceptional performance. Following standard GNN design principles suggested by You, Ying, and Leskovec³⁶, we use a Parameterized ReLU activation function³⁷ and a Batch Normalization layer³⁸ between graph convolutional layers within the GNN. These previous steps work to embed the nodes of each molecular graph, and to produce a graph-level embedding, we use a Self-Attention Graph Pooling mechanism^{39,40}.

We train two GNN blocks, one to embed the molecular structure of acids (Φ_a) and one to embed the molecular structure of glycols (Φ_g). These GNN components share an identical architecture. We present two ways of using these separate GNN blocks. The first considers training of both Φ_a and Φ_g with the same weights, updating them simultaneously within the model. In the second approach, each block is trained separately where weights are not shared across each of the models. Intuitively, this corresponds to learning two different models that embed acids and glycols in a way that is more advantageous for the downstream prediction network.

We obtain sets of molecular embeddings $\mathcal{A}_z = \{\mathbf{z}_1^a, \dots, \mathbf{z}_n^a \mid \mathbf{z}_i^a \in \mathbb{R}^d\}$ and $\mathcal{G}_z = \{\mathbf{z}_1^g, \dots, \mathbf{z}_m^g \mid \mathbf{z}_i^g \in \mathbb{R}^d\}$ for the acids and glycols, respectively, with the size of each set n, m varying with each input

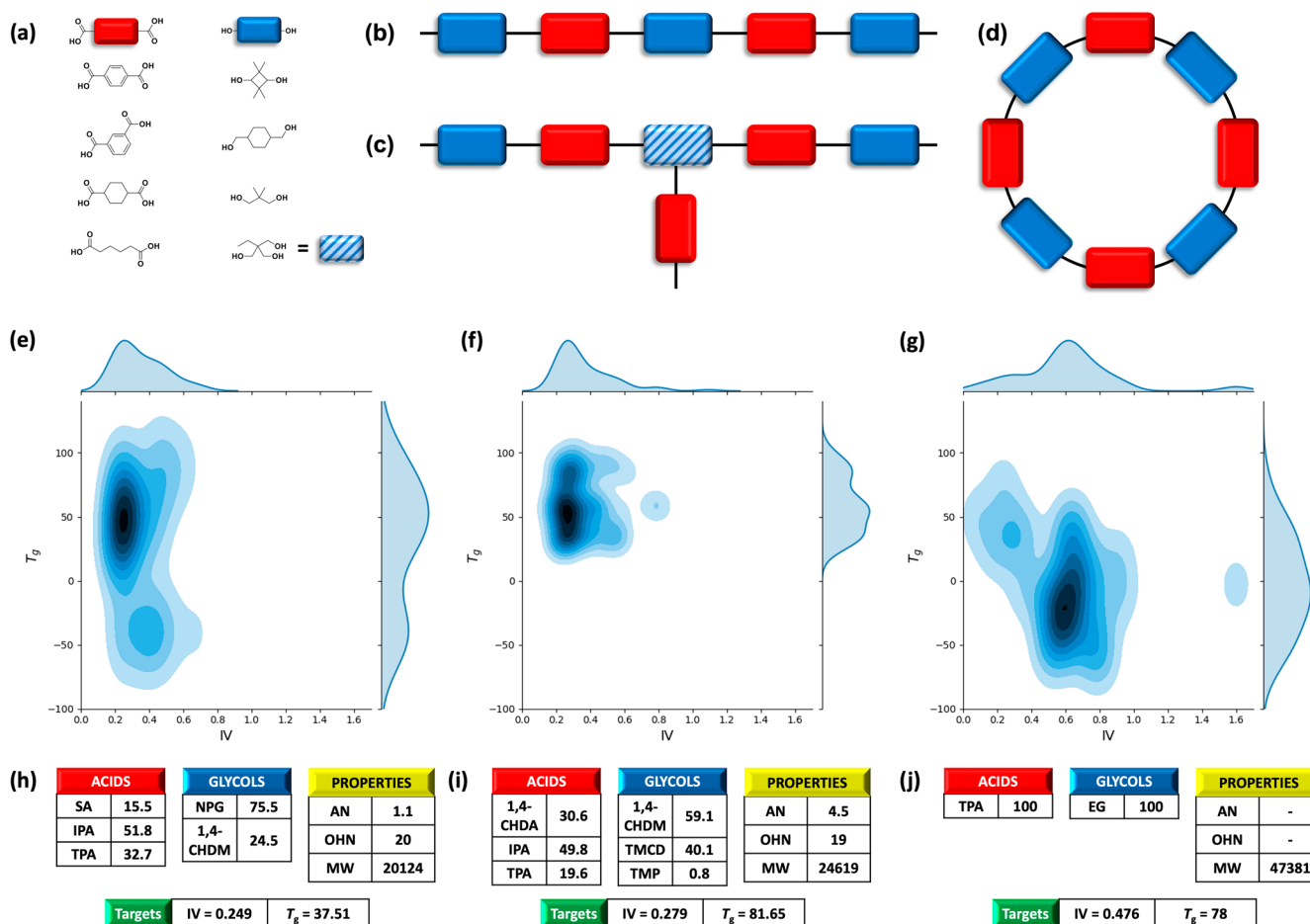


Fig. 1 The polyester database. **a** Polyesters are composed by combinations of diacid (red) and glycol (blue) monomers, and they can form **b** linear, **c** branched, and **d** cyclic chains that are present in linear polyesters. For branched polyesters, a small amount of trimethylolpropane (TMP, blue/white striped monomer) is required. The distribution of T_g and experimentally refined IV values for the **e** linear, **f** branched, and **g** “cyclic” polyesters demonstrate the heterogeneity of the total database. Representative examples of input/output values of a **h** linear, **i** branched, and **j** “cyclic” polyester. Each sample consists of a set of acid and glycol monomers together with their corresponding percentage, and a vector of resin properties: end-group statistics (AN and OHN) and weight-average molecular weight (M_w). T_g in °C, IV in dL/g, M_w in g/mol.

sample. Both sets of acids and glycols are permutation invariant, i.e., the ordering within each set is arbitrary.

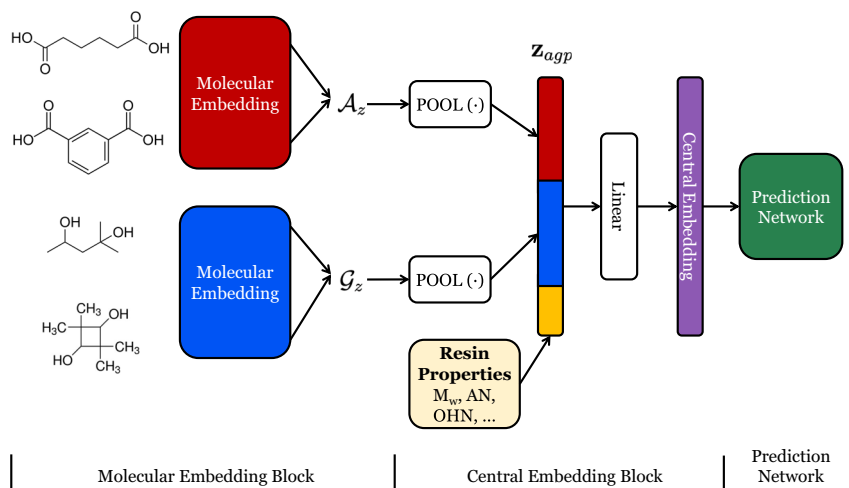
While the GNN shows the best performance in subsequent experiments with the proposed architecture, we note that any type of molecular embedding technique can be used in this pipeline, as long as the output is a one-dimensional vector of constant size. Therefore, this model can be easily amended to future developments in molecular representations, including more advanced GNN architectures. The advantage of the GNN embedding tool over deterministic molecular fingerprinting methods is that the model can be trained in an end-to-end fashion, i.e., the molecular representations can be tuned to different tasks and datasets.

Central embedding block

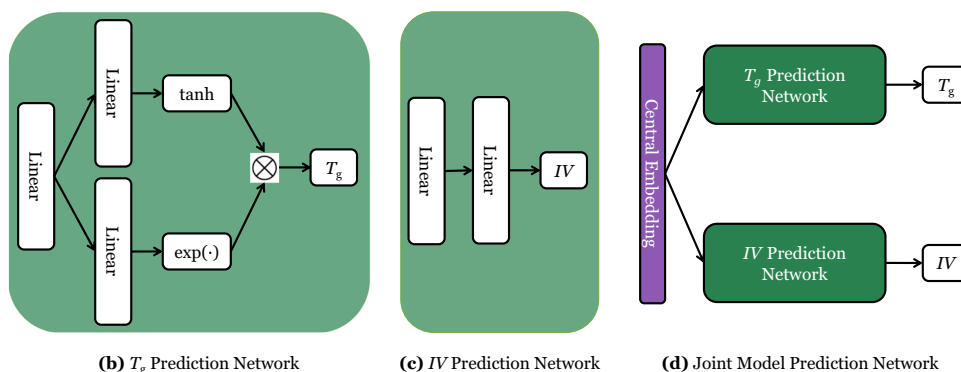
The central embedding block of POLYMERGNN combines all molecular embeddings into a chemically informed, constant-size vector for use in downstream tasks. The output \mathcal{A}_z and \mathcal{G}_z sets from the GNN blocks are permutation invariant sets of variable length. Thus, combining the embeddings requires a permutation-invariant aggregating function, or pooling function POOL(.). Some examples of these pooling operations are the element-wise SUM, MAX, or MEAN. We choose to use an element-wise MAX pooling in POLYMERGNN for predicting T_g and IV as early experiments showed

slight gains in performance with this pooling method. Applying the POOL(.) function to both \mathcal{A}_z and \mathcal{G}_z produces constant-size output vectors denoted as \mathbf{z}_a and \mathbf{z}_g .

The final portion of the central embedding layer incorporates the resin properties. Three key resin properties that are characteristic of polyesters were encoded in POLYMERGNN in addition to the structural information of each monomer. The considered properties are the weight-average molecular weight (M_w), the terminal acid number (AN), and terminal hydroxyl number (OHN) of the polymer chains. An additional property considered is that of explicit percentage of TMP that facilitates the synthesis of branched polymers. The introduction of branching can significantly change the shape of the polymer architecture since increasing the level of branching agent (TMP in this case) causes M_w to build more rapidly than the average molecular weight M_n as a function of reaction progress (decreasing OHN and AN), which leads to significant difference in the polydispersity index and IV^{41,42}. The TMP percentage is relevant for nearly half of the dataset and ranges from 0.0% (linear chains) to 15.7%. Explicitly providing TMP in the input gives the model a direct method to account for the approximate amount of branching in the final product of the synthesis. We will generally denote resin properties as $\mathbf{p} \in \mathbb{R}^{n \times m}$, for n samples in the dataset and m properties. The resin properties for sample i are denoted as $\mathbf{p}_i \in \mathbb{R}^m$. Denoting \oplus



(a) PolymerGNN Architecture

(b) T_g Prediction Network

(c) IV Prediction Network

(d) Joint Model Prediction Network

Fig. 2 Model architecture for POLYMERGNN. **a** The three major sections of the POLYMERGNN architecture are: (1) the molecular embedding blocks, (2) the central embedding mechanism, and (3) the prediction network. **b** Architecture for the T_g prediction network, where \tanh and $\exp(\cdot)$ correspond to a hyperbolic tangent activation function and the exponential function e^x , respectively, while \otimes corresponds to a multiplication of scalar outputs from \tanh and $\exp(\cdot)$. **c** Architecture for the IV prediction network. **d** Joint model setup, with “ T_g Prediction Network” and “IV Prediction Network” corresponding to the architectures shown in **b** and **c**, respectively.

as the concatenation operator, we construct one vector $\mathbf{z}_{\text{agp}} = \mathbf{z}_a \oplus \mathbf{z}_g \oplus \mathbf{p}_i$ for sample i . Note that this vector \mathbf{z}_{agp} is a constant size, as all constituent vectors composing it are also of constant size. We then use a fully connected neural network layer to transform this \mathbf{z}_{agp} into a central embedding vector, $\mathbf{z}_{\text{central}}$, which is enriched with information from the acid embeddings, glycol embeddings and resin properties of the given sample. This vector serves as input to downstream prediction models. In addition to resin properties, additional information can be added in the central embedding block, such as experimental data related to the differential scanning calorimetry parameters, the quenching rate used for measuring the glass transition temperature T_g , or the temperature under which IV values were obtained. Since a constant rate and a fixed temperature of 25° were applied for the T_g and IV measurements, respectively, our data are independent of the experimental conditions.

Prediction network

The prediction network predicts a given target value (T_g and/or IV) from the central embedding vector. We have explored whether it is advantageous to predict each property of interest separately (Fig. 2b, c for T_g and IV and, respectively) or both in a jointly trained model (Fig. 2d).

The prediction network for T_g consists of two separate branches, the prediction branch and the multiplier branch (Fig. 2b). In the prediction branch, the model uses a two-layer neural network to

learn an output T_g value, transforming the output with an exponentiation. This exponentiation is motivated by observations of a log-log relationship between some of the resin properties such as T_g and M_w as well as results from the ablation study (see Supplementary Note 5).

The prediction network for IV is a simple two-layer neural network with PReLU activation functions (Fig. 2c). We experiment with the same log-log transformation applied to the T_g network, as inspired by the Mark-Houwink Equation⁴³ relating M_w to IV. However, it was found through ablation studies that log-log transformation of input data and model output decreased performance of the IV model, so we use standard scaling of resin properties to produce the best results.

Finally, the joint model is trained to predict simultaneously both target values and shares similarities with the individual models. The difference from single-task models lies after the pooling and concatenation operation. After applying a linear layer and a PReLU activation function³⁷, the network diverges into two prediction branches—the T_g and IV branch. Each branch adopts an identical architecture to the prediction networks for T_g and IV.

Model behavior and performance

We experiment with replacing the model’s molecular embedding layer with several types of molecular representations, the simplest of which involves application of one resin property (M_w) to predict another property (T_g or IV). In addition, we encode the

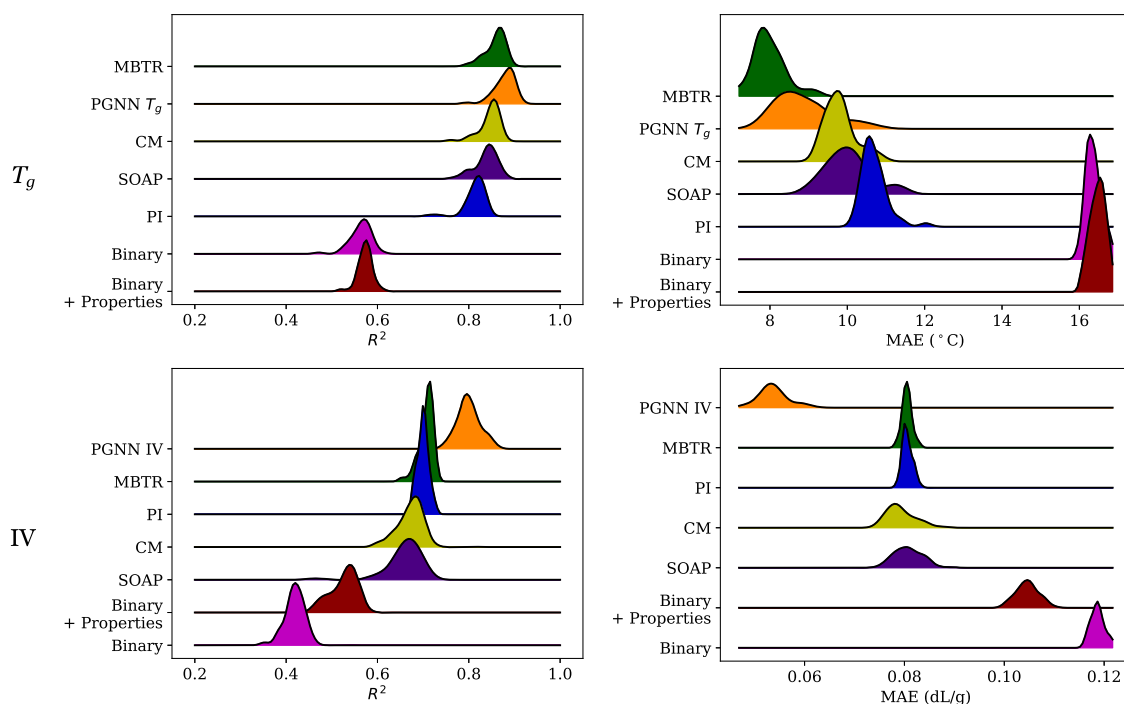


Fig. 3 Ridgeline plots of performance comparisons across different models for polymer property prediction. T_g results are shown on the top row while IV results are shown on bottom. In these trials, each task is a singular output, thus the top row shows results predicting only T_g while the bottom row shows results predicting only IV. The two left plots show the R^2 scores while the two right plots show the mean absolute error. Results are sorted by lowest mean MAE across each prediction task. Supplementary Table 13 shows the numerical results of these model comparisons. “PGNN” is an abbreviated notation for POLYMERGNN model.

composition of each resin using a binary approach. This is done by placing a 1 in the input vector at the location of each monomer (acid or glycol) that is present and a 0 in the input vector if the monomer is not present. This forms a vector of length twenty-five (thirteen different acids and twelve different glycols), which can be used as an identification of the specific resin in the dataset. To augment this approach, we have also added the M_w value at the end of this input vector to analyze its effect on the model accuracy. These two approaches are referred to as the ‘Binary’ method and the ‘Binary + Properties’ method, respectively.

Four additional molecular representations were tested, Coulomb Matrices (CM)⁴⁴, Smooth overlap of atomic potentials (SOAP)⁴⁵, Persistence Images (PI)⁴⁶, and Many-body Tensor Representations (MBTR)⁴⁷, which are popular non-deep learning methods to vectorize molecular structures. To keep comparisons similar to POLYMERGNN trials, we use the same resin features for each respective task that were found to be optimal for prediction with POLYMERGNN (see Methods). A kernel ridge regression algorithm is used to predict values for CM, SOAP, PI, and MBTR, as this was found to be the optimal model for learning on these representations in our previous wide-scale analysis.

Figure 3 shows a comparison of distributions of performance metrics across 50 trials of 5-fold cross validation on the dataset using the previously mentioned methods. POLYMERGNN outperforms other methods, yielding a higher R^2 score for both T_g and IV prediction tasks and an approximate 0.25 dL/g lower mean absolute error (MAE) when predicting IV. MBTR yields the next-best performance, even outperforming POLYMERGNN in MAE for the T_g prediction task. The distribution of POLYMERGNN metrics are wider—have a higher standard deviation—than for other approaches such as CM and MBTR. This is because the training of neural networks is more unstable under cross validation than a method such as kernel ridge regression, which is used to predict values for the other representations.

In order to test the joint prediction model, the molecular representations described in the previous experiments for T_g and IV model comparison were applied and their performance was evaluated. This modification replaced the “Molecular Embedding” block in Fig. 2 with different molecular representations. The downstream architecture of the joint prediction model was held equal in order to maintain the joint prediction task across each trial. The results are shown in Fig. 4. The POLYMERGNN model ultimately outperforms other methods across both tasks, producing the lowest R^2 and MAE errors for both T_g and IV prediction tasks. However, several embedding techniques perform well on this task, especially MBTR, demonstrating that the proposed model architecture can sufficiently learn both T_g and IV jointly with multiple types of molecular embedding techniques.

Computational screening of polymers

In order to demonstrate the applicability of POLYMERGNN, we have screened a virtual database of 1000 materials with variable compositions. We chose isophthalic acid (IPA), terephthalic acid (TPA), adipic acid (AA), 2-methyl-1,3-propanediol (MP Diol), and 1,4-cyclohexanedimethanol (1,4-CHDM) due to their widespread use in polyester materials. In addition, we varied the OHN value in the input vector while the AN was kept fixed (value of 1), effectively varying the molecular weight by changing the stoichiometry of diacids and diols. We train a joint POLYMERGNN instance on the entire labeled dataset described in section “Results”; this model is then used to predict T_g and IV values on the large virtual database. This procedure is performed ten separate times on the same dataset in order to provide confidence levels for each sample in the set.

In Fig. 5a, we plot the results of our screening analysis for T_g and IV predictions. This plot shows the strong correlation between M_w and IV prediction, which is expected based on classical relationships⁴³. Interestingly, POLYMERGNN identifies several candidates falling into the high- T_g , low-IV region (top left) of the plot, a

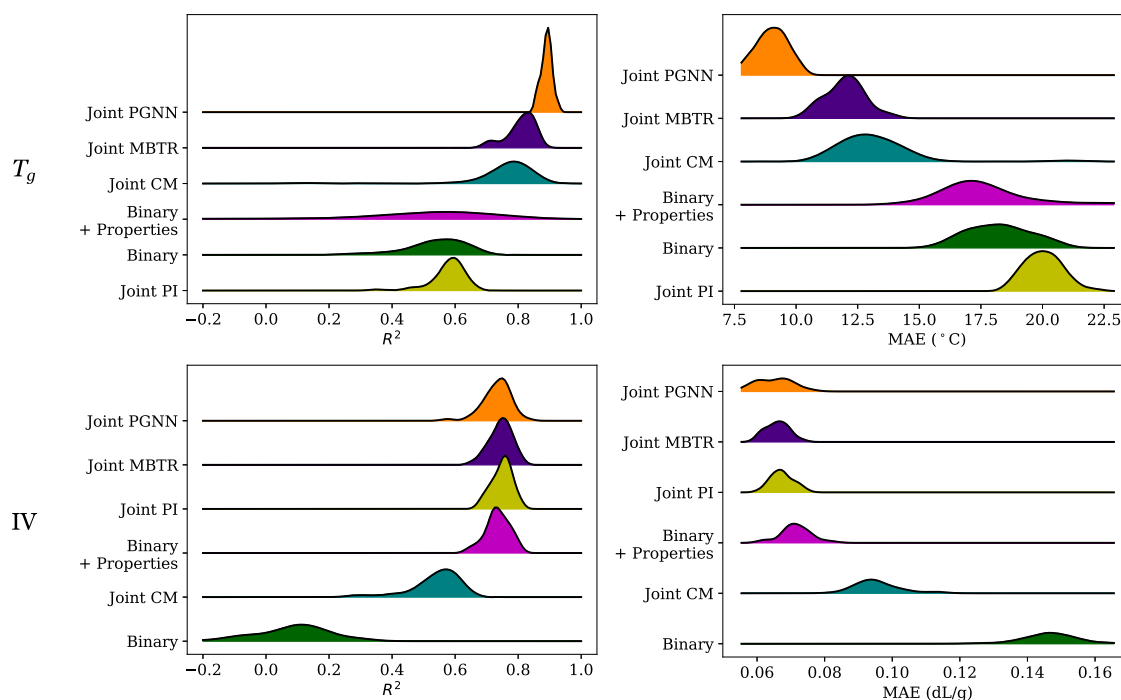


Fig. 4 Ridgeline plots of performance comparisons across multiple representations used for the joint learning task of T_g and IV. T_g results are shown on the top row while IV results are shown on bottom. The two left plots show the R^2 scores while the two right plots expose the mean absolute error. Results are sorted by lowest mean MAE across each prediction task. Supplementary Table 14 shows the numerical results of these model comparisons. “PGNN” is an abbreviated notation for POLYMERGNN model.

region that could be of interest to some niche applications. We provide additional information on the composition of these interesting polymers along with other experimental details in Supplementary Note 7.

Figure 5b shows the inverse relationship between adipic acid and T_g . As the concentration of more flexible monomers such as adipic acid increase in the composition, the polymer backbone will require less thermal energy to move around and thus will resist forming glasses at lower temperatures. Increasing the concentration of more rigid, stiffer components will have the opposite effect. A small region of the plot seems to contradict this largely negative correlation, namely the polymers above 60% adipic acid; these polymers seem to increase in T_g as the percentage of adipic acid increases. However, these samples have higher standard error relative to the entire plot. While M_w distributions seem to be the same, the OHN values are slightly lower on average for the outliers (see Supplementary Note 7). This discrepancy might be causing some out-of-distribution effects since OHN typically directly correlates with M_w . It is reasonable to conclude that these samples were simply out-of-distribution from the original training data, thus causing the model to predict outside of the expected relationship between adipic acid and T_g . This shows the utility of using standard error as an uncertainty statistic for predictions in the screen.

Explainability

We examine the attribution scores given to the resin properties for a given material using the Grad-CAM attribution method⁴⁸. Attribution scores in this context can be interpreted as the relative importance of all variables, with more positive values indicating the highest importance.

In the T_g plot (Fig. 6a), we see that M_w is important for the prediction, but less important than having information on the molecular structure of the acids and glycols. For the IV prediction, M_w has the largest overall attribution of all variables, including acid and glycol embeddings (Fig. 6b). As a result, it is reasonable

to conclude that M_w is very important for predicting IV, which matches chemical intuition based on the Mark-Houwink equation⁴³ directly relating IV to M_w and the strong correlation seen in M_w and IV predictions in the computational screening. AN, OHN, and TMP seem to have less of an importance in predicting IV values, which mirrors results seen in the ablation study. This also highlights the fact that although these parameters can be used to calculate a theoretical M_w ^{41,42}, additional variables that are difficult to experimentally capture in complex copolymer compositions must be considered (i.e., the presence of additional end-groups beyond COOH and OH and non-statistical distributions of monomers throughout the polymer backbone). Finally, both acid and glycol embeddings are shown to have great importance for both prediction tasks. Glycol embeddings are slightly more important than acid embeddings in the IV prediction task, but both acid and glycol embeddings seem to be equally important for T_g prediction.

DISCUSSION

This work proposes POLYMERGNN, a general framework, GNN-based machine learning model for single-task and multitask learning of polymer properties. POLYMERGNN uses as input a graph-based representation of each monomer present in a material, and it is able to provide high accuracy for predicting polymer properties. The model can embed and process an arbitrary number of input monomers as sets that are permutation-invariant, i.e., the order of molecular inputs is not relevant. In addition, POLYMERGNN computes embeddings that are useful in downstream tasks. This benefit is demonstrated in the joint POLYMERGNN model, where a model is trained to predict both T_g and IV with performance metrics close to that of the model trained on a single task. Because of superior joint prediction performance, representations learned by the model may be transferable to differing downstream tasks. Therefore, POLYMERGNN could potentially learn properties on which

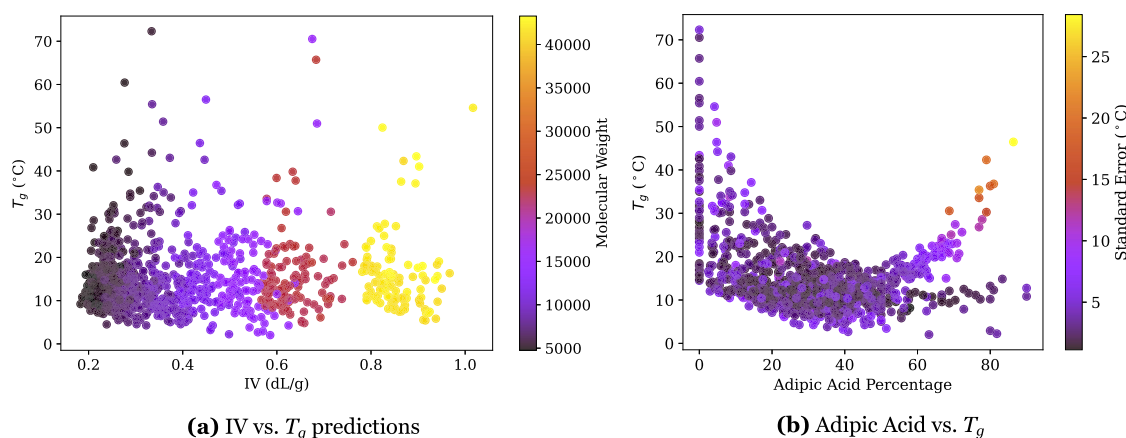


Fig. 5 Results of a large-scale screen of POLYMERGNN on a computationally generated dataset. **a** is colored by M_w to accentuate the strong positive correlation of M_w and IV learned by the model. **b** shows how adipic acid is negatively correlated to T_g ; points are colored by standard error to highlight a low-confidence region in the high-adipic acid, high- T_g scenario.

there is limited data (*few-shot learning*) by using embeddings from a model pre-trained on another task with more abundant data.

For this project, more than 240 polyesters were synthesized and their properties such as glass transition temperature (T_g) and intrinsic viscosity (IV) were compiled in a database. POLYMERGNN demonstrated remarkable accuracy for both properties, independently if it was trained on one or both properties. In addition, combination of the POLYMERGNN architecture with other commonly used molecular representations managed to provide increased performance when compared to models that do not use the architecture that was developed for this project. Further development of POLYMERGNN can include the use of self-attention mechanisms⁴⁹, a useful approach to encode dependencies between monomer inputs. Finally, this type of design is not fixed to polyesters, as is described in this work, but can rather be transferred to the prediction of other types of polymers and properties.

METHODS

Polyester resin synthesis

The polyols were produced using either a resin kettle reactor setup via solvent-assisted polycondensation or a resin rig reactor setup via melt polycondensation, both of which were controlled with automated control software.

The solvent-assisted resins were produced on a 3.5 mole scale using a 2 L kettle with overhead stirring and a partial condenser topped with total condenser and Dean Stark trap. Approximately 10 wt% (based on reaction yield) azeotroping solvent of high boiling point (A150 and A150ND) was used to both encourage egress of the water condensate out of the reaction mixture and keep the reaction mixture viscosity at a reasonable level using the standard paddle stirrer. Chemical reagents were added to the kettle, which was then completely assembled. The Fascat 4100 (monobutyltin oxide) catalyst was added via the sampling port after the reactor had been assembled and blanketed with nitrogen for the reaction. Additional A150/A150ND solvent was added to the Dean Stark trap to maintain the ~10 wt% solvent level in the reaction kettle. The reaction mixture was heated without stirring from room temperature to 150 °C using a set output controlled through the automation system. Once the reaction mixture was fluid enough, the stirring was started to encourage even heating of the mixture. At 150 °C, the control of heating was switched to automated control and the temperature was ramped to 230 °C over the course of 4 h. The reaction was held at 230 °C for 1 h and then heated to 240 °C over the course of 1 h. The reaction was then held at 240 °C and sampled every 1–2 h

upon clearing until the desired acid value was reached. The ~90%-solids resins were ground into 6 mm pellets and thoroughly dried in a vacuum oven at 150 °C for 24 h prior to characterization.

The melt polycondensation resins were produced on a 0.5 mole scale. A 500 mL, one-neck, round-bottom flask was carefully charged with all chemical reagents and Fascat 4100 (monobutyltin oxide) catalyst. The flask was equipped with a polymer head adapter with stainless steel mechanical stirrer and securely clamped to the polymerization rig. To the polymer head, a distillation side arm and Erlenmeyer flask were attached. The automated-controlled vacuum system was attached to the flask side arm to allow for a reduction in pressure of the reaction vessel. The Belmont metal bath was preheated to 20 °C above the recipe starting temperature (180 °C). The apparatus was subjected to two iterations of a nitrogen (N_2) purge to remove oxygen and then dunked into the metal bath to begin. The flask was held at 180 °C for 10 min to melt the starting materials and then stirring was started to encourage even heating on the mixture. The flask was heated to 240 °C over 4 h and then held there for an additional hour. Pressure in the reaction flask was reduced to 1.5 torr over 45 min and then the reaction was held at 1.5 torr until the final acid value was reached. Dry-ice was used to ensure that the solvent traps were sufficiently cold to prevent any solvent/organic matter from going to the vacuum pump. After completion, the flask was slowly brought back to atmospheric pressure and removed from the hot metal bath. Upon solidification, the polymer was pulled from the round-bottom flask by partially melting the edges, then the glass flask was broken with a hammer to give the solid polymer 'lollipop' on the stir rod. The polymer was cooled in dry-ice, removed from the stir-rod, and ground into 6 mm pellets prior to characterization.

Polyester resin characterization

Acid number (AN) was determined using colorimetric titration in pyridine with phenolphthalein indicator and 0.1 N KOH titrant administered with an auto-dispensing titrator. Hydroxyl number (OHN) was determined via 1H NMR end-group analysis on a Bruker 500 MHz spectrometer or by reaction of hydroxyl groups with p-toluenesulfonyl isocyanate and subsequent potentiometric titration of the acid carbamate product. The OHN results obtained are then corrected for contributing acid number. The inherent viscosity (IV) of all polymers was determined in 0.5 wt% PM 95 (60/40 phenol/1,1,2,2-tetrachloroethane) solution at 25 °C. Molecular weights were determined by gel permeation chromatography (GPC) with 95/5 methylene chloride/HFIP mobile phase and calibration curves for polystyrene standards. Monomer

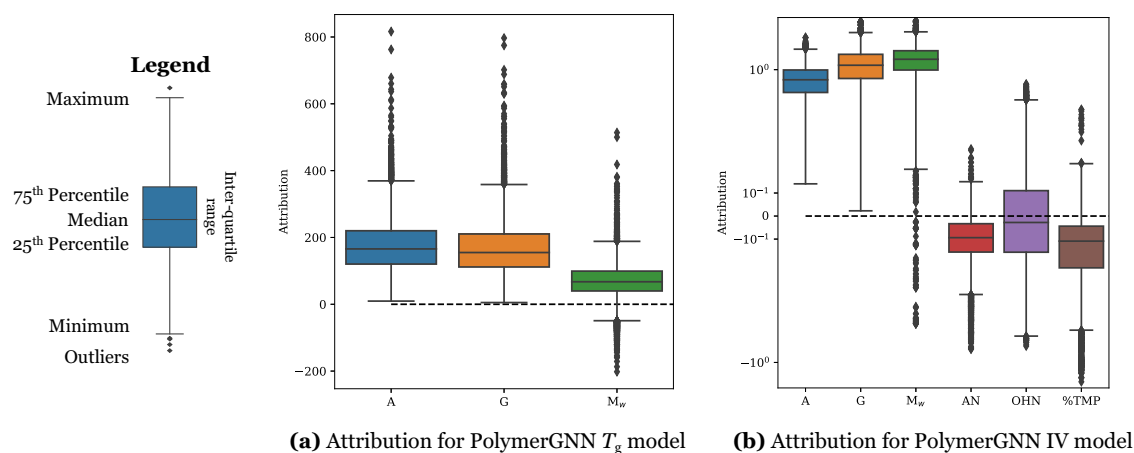


Fig. 6 Attribution scores. POLYMERGNN T_g model **(a)** and POLYMERGNN IV model **(b)** attribution scores computed by Grad-CAM on the central embedding layer of the model. A log-scale is used in **b** to show the separation between components with small attribution scores. Distributions are shown for attributions from every trained model for 50 5-fold cross validations on the dataset.

composition was determined via gas chromatography (GC) hydrolysis. The glass transition temperature (T_g) was determined using differential scanning calorimeter (DSC) at 20 °C/min. ramp rate with a N_2 sweep. The T_g was based on second heat thermograms.

Linear polyesters typically have a polydispersity index (PDI) between 1.5–2.5 and branched polyesters can have a much wider range depending on how much branching agent is added, the degree of polymerization, and other factors. Our polyester dataset includes both linear and branched resins. IV is affected by both resin composition and molecular weight, with M_w more strongly affecting IV than the average molecular weight M_n . As M_w increases, typically so does IV. In our dataset, increased PDI is usually associated with increased M_w due to branching and this consequently increases the IV. Similarly, T_g increases as M_w and IV increase and begins to plateau at higher IVs and M_w s. These relationships can be quantified for specific polyester compositions, but becomes very complicated in complex datasets such as described in this paper and therefore machine learning can help make useful predictions in this design space.

Quantum chemical calculations

The individual monomers for each resin were optimized using metadynamics to sample the conformational space of the monomer. The Conformer-Rotamer Ensemble Sampling Tool (CREST)⁵⁰ was utilized in order to sample the conformational space of each monomer and generate a list of minimum energy conformation using semiempirical density functional tight binding (DFTB). Tight convergence criteria was utilized for both the geometry optimizations and the self-consistent-field cycles of DFTB. The lowest energy conformer generated was then used as the input structure for the different molecular representations.

Graph neural network

We will establish some preliminaries for graph neural networks. Let $G=(V,E)$ be a graph with V nodes and E edges. If G is a molecular graph, we consider V to consist of all atoms in the molecule, and E to comprise all bonds between those atoms. Indeed, if v_i and v_j are atoms in V , a bond between them would be denoted by the edge $(v_i, v_j) \in E$. It is also useful to define the *neighborhood* of a node v_i , denoted $\mathcal{N}(v_i)$. The neighborhood is the set of all nodes for which there exists an edge connecting to v_i , i.e., $\mathcal{N}(v_i) = \{v_j | (v_i, v_j) \in E\}$. For each $v_i \in V$, we have a d -dimensional feature x_i . The collection of node features for all n nodes in a graph is denoted by $\mathcal{X} = \{x_1, \dots, x_n\}$, $x_i \in \mathbb{R}^d$ and may

include atomic properties such as atomic charge, atomic mass, or scalar properties associated with each atom in the molecule. In POLYMERGNN, we use six properties: the charge, degree, mass, aromaticity—a Boolean variable indicating whether the atom is found within an aromatic portion of the molecule—the explicit number of hydrogen atoms bonded with the atom, and the number of valence electrons. All features are extracted automatically using RDKit⁵¹. In a similar manner to node features, edge features can also be introduced into the graph construction; however, we omit any additional edge features in this work since preliminary benchmarking showed no empirical boost in performance. This described formulation allows us to treat molecular representations as a graph onto which we can apply graph machine learning algorithms and methods, namely graph neural networks.

A graph neural network (GNN) is a machine learning algorithm that learns embeddings of nodes within a graph. These so-called node-level embeddings can be combined into a graph-level embedding that represents the entire graph G . Graph-level embeddings can be used in downstream prediction tasks, such as predicting T_g or IV. We specifically focus on graph convolutional neural networks, thus when mentioning the term “GNN” in this work, it is assumed that a graph convolutional neural network is being discussed.

A GNN model consists of iterations of AGGREGATE-COMBINE steps that update the representation of nodes by aggregating information from the local topology of the graph. We denote $h_i^{(l)}$ as the representation for node v_i at layer l of the network. As an initial step, we set $h_i^{(0)} = x_i$. Each layer l then performs the following functions to obtain the successive layer’s node embeddings: $a_i^{(l)} = \text{AGGREGATE}^{(l)}(\{h_j^{(l-1)} | v_j \in \mathcal{N}(v_i)\})$, such that $h_i^{(l)} = \text{COMBINE}^{(l)}(h_i^{(l-1)}, a_i^{(l)})$. Intuitively, the AGGREGATE and COMBINE functions work to mix information between neighboring atoms within the molecule. Different GNN layers introduce variations to the AGGREGATE and COMBINE functions. Common functions for the AGGREGATE step are MEAN and MAX while COMBINE is commonly performed by a single, fully connected neural network, as in refs. 52,34, and ref. 35. To produce a graph-level embedding, h_G , a READOUT function is used to pool all node representations from the graph, i.e., $h_G = \text{READOUT}(\{h_i^{(L)} | v_i \in V\})$. After the READOUT operation, it is guaranteed that h_G is a constant-size vector regardless of the size of G . READOUT can be performed by simple, permutation-invariant function such as MEAN, MAX, or more advanced pooling methods^{39,53}. In this work, we utilize the self-attention pooling mechanism³⁹. After the READOUT function is performed, the resulting h_G should contain

information about the entire graph in question, making this representation useful in downstream prediction tasks.

Loss function

Both T_g and IV tasks utilize the Mean Squared Error (MSE) loss function to train the networks. The joint model was trained using the following loss function, $L = \gamma L_{IV} + L_{T_g}$, where L_{IV} is the MSE of the IV prediction with respect to the true IV value and L_{T_g} is the MSE of the T_g prediction with respect to the true T_g value. The γ constant serves as a weighing factor to scale the IV loss in proportion to how much the models should prioritize learning IV relevant features. The IV and T_g learning tasks have different units that have varying scales. The scale of T_g values is much larger than that of IV; this would result in MSE being very large for T_g while the MSE is very low for IV, even if the performance is equal for each of them. Therefore, we set γ to a very large arbitrary value (10,000 herein) to offset the effect of units for this joint learning problem. Another rationale for a large γ is because it prioritizes the IV task, which is more difficult to learn based on previous trials.

DATA AVAILABILITY

All DFTB-optimized geometries are available in the Supplementary Information.

CODE AVAILABILITY

We provide all code for POLYMERGNN in our open-source GitHub repository, which can be found at <https://anonymous.4open.science/r/PolymerGNN-6736>.

Received: 11 October 2022; Accepted: 3 May 2023;

Published online: 30 May 2023

REFERENCES

1. Tsui, A., Wright, Z. C. & Frank, C. W. Biodegradable polyesters from renewable resources. *Annu. Rev. Chem. Biomol. Eng.* **4**, 143–170 (2013).
2. Alagirusamy, R. & Das, A. In *Polyesters and Polyamides*. 219–252 (Woodhead Publishing Limited, 2008).
3. Fangueiro, R., Pereira, C. G. & Araujo, M. D. In *Polyesters and Polyamides*. 542–592 (Wiley-VCH, 2008).
4. Militky, J. In *Handbook of Tensile Properties of Textile and Technical Fibres*. 223–314 (Woodhead Publishing Limited, 2009).
5. Gromski, P. S., Henson, A. B., Granda, J. M. & Cronin, L. How to explore chemical space using algorithms and automation. *Nat. Rev. Chem.* **3**, 119–128 (2019).
6. Reymond, J. L., Van Deursen, R., Blum, L. C. & Ruddigkeit, L. Chemical space as a source for new drugs. *MedChemComm* **1**, 30–38 (2010).
7. Reymond, J. L. The chemical space project. *Acc. Chem. Res.* **48**, 722–730 (2015).
8. Wu, S. et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *npj Comput. Mater.* **5**, 66 (2019).
9. Molina, J., Laroche, A., Richard, J.-V., Schuller, A.-S. & Rolando, C. Neural networks are promising tools for the prediction of the viscosity of unsaturated polyester resins. *Front. Chem.* **7**, 375 (2019).
10. Kumar, J. N. et al. Machine learning enables polymer cloud-point engineering via inverse design. *npj Comput. Mater.* **5**, 73 (2019).
11. Li, Z., Hao, K., Chen, L., Ding, Y. & Huang, B. Pet viscosity prediction using jit-based extreme learning machine. *IFAC-PapersOnLine* **51**, 608–613 (2018).
12. Rizkin, B. A. & Hartman, R. L. Supervised machine learning for prediction of zirconocene-catalyzed alpha-olefin polymerization. *Chem. Eng. Sci.* **210**, 115224 (2019).
13. Doan Tran, H. et al. Machine-learning predictions of polymer properties with polymer genome. *J. Appl. Phys.* **128**, 171104 (2020).
14. Tao, L., Varshney, V. & Li, Y. Benchmarking machine learning models for polymer informatics: an example of glass transition temperature. *J. Chem. Inf. Model.* **61**, 5395–5413 (2021).
15. Li, H. et al. Tuning the molecular weight distribution from atom transfer radical polymerization using deep reinforcement learning. *Mol. Syst. Des. Eng.* **3**, 496–508 (2018).
16. Zhang, Y. & Xu, X. Machine learning glass transition temperature of polyacrylamides using quantum chemical descriptors. *Polym. Chem.* **12**, 843–851 (2021).

17. Zhang, Y. & Xu, X. Machine learning glass transition temperature of polymers. *Heliyon* **6**, 1–7 (2020).
18. Zhang, Y. & Xu, X. Machine learning glass transition temperature of poly-methacrylates. *Mol. Cryst. Liq. Cryst.* **730**, 9–22 (2021).
19. Alcobaca, E. et al. Explainable machine learning algorithms for predicting glass transition temperatures. *Acta Mater.* **188**, 92–100 (2020).
20. Tao, L., Chen, G. & Li, Y. Machine learning discovery of high-temperature polymers. *Patterns* **2**, 100225 (2021).
21. van Krevelen, D. W. & te Nijenhuis, K. *Properties of Polymers* (Elsevier, 2009).
22. Park, J. et al. Prediction and interpretation of polymer properties using the graph convolutional network. *ACS Polym. Au* **2**, 213–222 (2022).
23. Zeng, M. et al. Graph convolutional neural networks for polymers property prediction. <https://arxiv.org/abs/1811.06231> (2018).
24. Dai, M., Demirel, M., Liang, Y. & Hu, J.-M. Graph neural networks for an accurate and interpretable prediction of the properties of polycrystalline materials. *npj Comput. Mater.* **7**, 103 (2021).
25. Fung, V., Zhang, J., Juarez, E. & Sumpter, B. G. Benchmarking graph neural networks for materials chemistry. *npj Comput. Mater.* **7**, 84 (2021).
26. Jiang, D. et al. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *J. Cheminf.* **13**, 12 (2021).
27. Wieder, O. et al. A compact review of molecular property prediction with graph neural networks. *Drug Discov. Today. Technol.* **37**, 1–12 (2020).
28. Deng, D. et al. Xgraphboost: extracting graph neural network-based features for a better prediction of molecular properties. *J. Chem. Inf. Model.* **61**, 2697–2705 (2021).
29. Deng, D. et al. Xgraphboost: extracting graph neural network-based features for a better prediction of molecular properties. *J. Chem. Inf. Model.* **61**, 2697–2705 (2021).
30. Ma, H. et al. Cross-dependent graph neural networks for molecular property prediction. *Bioinformatics* **38**, 2003–2009 (2022).
31. Lin, T.-S. et al. Bigsmiles: a structurally-based line notation for describing macromolecules. *ACS Cent. Sci.* **5**, 1523–1531 (2019).
32. Lin, T.-S., Rebello, N. J., Lee, G.-H., Morris, M. A. & Olsen, B. D. Canonicalizing bigsmiles for polymers with defined backbones. *ACS Polym. Au* **2**, 486–500 (2022).
33. Aldeghi, M. & Coley, C. W. A graph representation of molecular ensembles for polymer property prediction. *Chem. Sci.* **13**, 10486–10498 (2022).
34. Veličković, P. et al. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings* (Vancouver Convention Center, Vancouver, BC, Canada, 2018).
35. Hamilton, W., Ying, Z. & Leskovec, J. Inductive representation learning on large graphs. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017). https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e99-Paper.pdf.
36. You, J., Ying, Z. & Leskovec, J. Design space for graph neural networks. In: *Advances in Neural Information Processing Systems* (2020). <https://proceedings.neurips.cc/paper/2020/file/c5c3d4fe6b2cc463c7d7ecba17cc9de7-Paper.pdf>.
37. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026–1034 (IEEE, 2015).
38. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning—Volume 37, ICML'15*, 448–456 (JMLR.org, 2015).
39. Lee, J., Lee, I. & Kang, J. Self-attention graph pooling. In *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, 3734–3743 (PMLR, 2019).
40. Knyazev, B., Taylor, G. W. & Amer, M. Understanding attention and generalization in graph neural networks. In: *Advances in Neural Information Processing Systems*, 4202–4212 (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.
41. Flory, P. J. Fundamental principles of condensation polymerization. *Chem. Rev.* **39**, 137–197 (1946).
42. Stockmayer, W. H. Molecular distribution in condensation polymers. *J. Polym. Sci.* **9**, 69–71 (1952).
43. Hiemenz, P. C. & Lodge, T. P. *Polymer Chemistry* (Taylor and Francis, 2007).
44. 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.* **108**, 058301 (2012).
45. Bartok, A. P., Kondor, R. & Csanyi, G. On representing chemical environments. *Phys. Rev. B* **87**, 219902 (2013).
46. Townsend, J., Micucci, C. P., Hymel, J. H., Maroulas, V. & Vogiatzis, K. D. Representation of molecular structures with consistent homology for machine learning applications in chemistry. *Nat. Commun.* **11**, 1–9 (2020).

47. Huo, H. & Rupp, M. Unified representation of molecules and crystals for machine learning. *Mach. Learn.: Sci. Technol.* **3**, 045017 (2022).
48. Selvaraju, R. R. et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, 618–626 (IEEE, 2017).
49. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations* (San Diego, CA, USA, 2015).
50. Pracht, P., Bohle, F. & Grimme, S. Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Phys. Chem. Chem. Phys.* **22**, 7169–7192 (2020).
51. Landrum, G. Rdkit: Open-source cheminformatics. <https://www.rdkit.org>. Accessed 1 January (2022).
52. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? In: *International Conference on Learning Representations* (New Orleans, Louisiana, USA, 2019).
53. Ying, Z. et al. Hierarchical graph representation learning with differentiable pooling. In: *Advances in Neural Information Processing Systems*, vol. 31 (NeurIPS Proceedings, 2018).

ACKNOWLEDGEMENTS

This research is generously supported by Eastman Chemical Company, grant no. EMN-20-F-S-01. We also acknowledge the Infrastructure for Scientific Applications and Advanced Computing (ISAAC) of the University of Tennessee.

AUTHOR CONTRIBUTIONS

B.P.A., C.L.B., V.M., and K.D.V. conceived the project. C.L.B. synthesized and characterized the polyesters, and collected all experimental data for the database formation. G.A.M. performed all quantum chemical calculations. O.Q., S.T., V.M., and K.D.V. conceptualize the POLYMERGNN approach. O.Q. and S.T. wrote and tested the POLYMERGNN code. O.Q., S.T., and G.A.M. trained multiple ML models for the evaluation of POLYMERGNN. All authors wrote the paper.

COMPETING INTERESTS

The authors declare no competing non-financial interests but the following competing financial interests: B.P.A. and C.L.B. are employees of Eastman. The experimental data used were provided by Eastman. All other declares no competing interest.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41524-023-01034-3>.

Correspondence and requests for materials should be addressed to Cameron L. Brown, Vasileios Maroulas or Konstantinos D. Vogiatzis.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023