Macromolecules

pubs.acs.org/Macromolecules Article

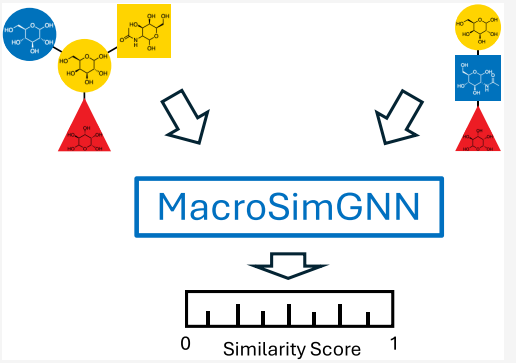
MacroSimGNN: Efficient and Accurate Prediction of Macromolecule Pairwise Similarity via A Graph Neural Network

Jiale Shi, Runzhong Wang, Nathan J. Rebello, Jiarui Lu, Bradley D. Olsen,* and Debra J. Audus*

Cite This: *Macromolecules* 2026, 59, 1885−1900 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Efficient and accurate prediction of macromolecule pairwise similarity is essential for developing database search engines and is useful for machine learning based predictive tools. Existing methods for calculating macromolecular similarity suffer from significant drawbacks. Graph edit distance is accurate but computationally expensive, and graph kernel methods are computationally efficient but inaccurate. This study introduces a graph neural network model, MacroSimGNN, which significantly improves computational efficiency while maintaining high accuracy on macromolecule pairwise similarity. Furthermore, this approach enables feature embeddings based on macromolecular similarities to a set of landmark molecules, enhancing both unsupervised and supervised learning tasks. This method represents a significant advancement in macromolecular cheminformatics, paving the way for the development of advanced search engines and data-driven design of macromolecules.

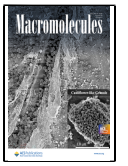


INTRODUCTION

Macromolecules are both ubiquitous and indispensable.[1,2] Biological macromolecules, such as glycans,[3,4] proteins[5−7] and nucleic acids,[8−10] are essential for life, serving as catalysts for survival and growth functions, while synthetic macromolecules find extensive use in fields such as textiles,[11] water purification,[12,13] energy,[14] transportation,[15] construction,[16] and biotechnology.[17] Macromolecule similarity offers insights into quantitative structure−property relationships[18,19] as similar macromolecules are more likely to have similar properties. Similarity is also essential for efficient search algorithms for macromolecule databases by enabling ranking of targets.[20−26] Furthermore, macromolecular similarity either forms the basis or enhances machine learning techniques, including clustering, classification, and regression for predicting properties and discovering new macromolecular materials.[19,27−44] Based on chemical intuition and experience, similar polymers or macromolecules are more likely to exhibit similar properties, and require similar synthetic strategies. When experimental researchers aim to design new polymers or estimate their physical properties, pairwise similarity searches can be useful for identifying previously synthesized polymers that resemble the target structures. This can assist in synthesis planning and provide a simple, rough estimate of the physical and chemical properties of the new polymers.

While sequence matching algorithms[45,46] can be used for similarity calculations in simple linear macromolecules, many complex macromolecules have nonlinear topologies.[4,47−50] To address this, both atomistic and coarse-grained graph representations[51−53] were developed for macromolecule similarity calculations. However, the graph similarity calculations between atomistic graph representations of macromolecules are computationally expensive and impractical due to the large number of atoms compared to small molecules. Consequently, coarse-grained graph representations[47,54,55] were utilized, where nodes represent monomers and edges represent connections between monomers. Two main approaches have been used to calculate pairwise similarity in these coarse-grained representations: graph edit distance (GED)[47,54,55] and graph kernels.[47,56−59] GED measures the minimum operation costs to transform one graph to another.[60] However, GED is a nondeterministic polynomial-time hardness (NP-hard) problem. Even with coarse-grained representations, computing the exact GED remains costly,[47,54,55,61,62] limiting its use in scale-up or time-sensitive applications. Graph kernel methods map graphs to a high-dimension space and measure similarity between graphs using inner products in that space. Graph kernels often provide an approximation of graph similarity rather than an exact measure since the mapping processing and inner product operation can lose small but important structural differences between graphs. Therefore, graph kernel methods offer improved efficiency but often suffer from reduced

Received: October 29, 2025
Revised: January 6, 2026
Accepted: January 29, 2026
Published: February 16, 2026



https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

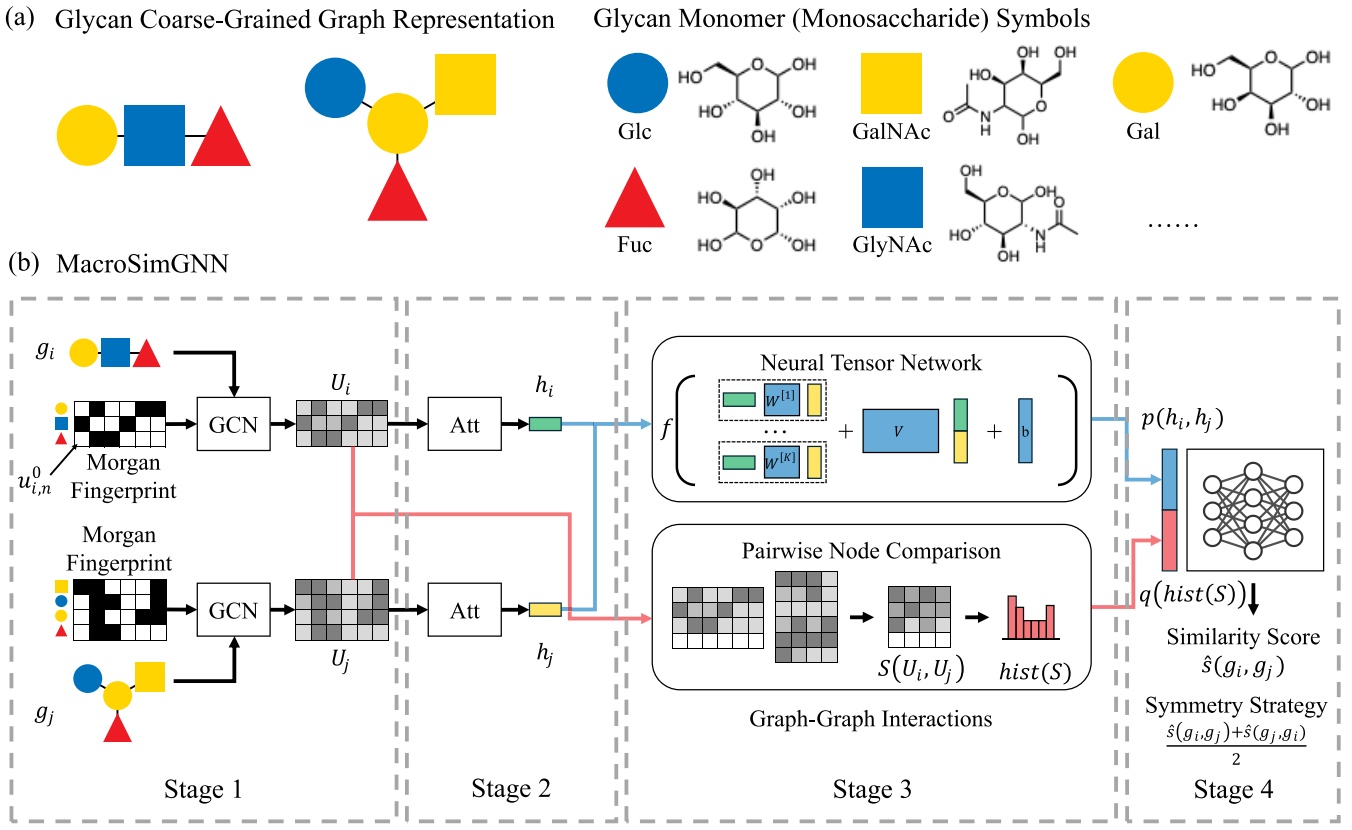


Figure 1. (a) Coarse-grained graph representations of glycans where the nodes represents glycan monomers (monosaccharides). (b) Schematic representation of the MacroSimGNN methodology. The four-stage process of MacroSimGNN, adapted from SimGNN.[62] Stage 1 includes graph convolutional networks (GCNs) for node-level embedding. Nodes represent monomers, which are embedded using Morgan molecular fingerprints, a key modification from SimGNN. Stage 2 is graph-level embedding with the global context-aware attention (Att) layer, which generates graph embedding vectors by aggregating node embeddings with learned weights. Stage 2 is for the neural tensor network used in Stage 3; it is not used for the pairwise node comparison in Stage 3. Stage 3 is for graph−graph interactions which include a neural tensor network and pairwise node comparison. Stage 4 is fully connected network layers for the prediction of similarity scores. At the end, a symmetry strategy is employed to ensure that the order of the pairs of graphs, g_i and g_j, do not matter. This adapted framework enables efficient macromolecular similarity predictions while maintaining computational efficiency.

accuracy.[47,56−59] Several recent advances in deep learning demonstrated that deep neural networks have the potential to learn graph matching-related tasks, leading to state-of-the-art matching accuracy while also benefits from the efficiency.[62−64]

Bai et al.[62] proposed the SimGNN framework which is a graph neural network approach designed for rapid and accurate computation of small molecule pairwise graph similarity. In SimGNN,[62] each small molecule is represented as a chemical compound graph, with nodes representing atoms and embedded using one-hot encoding. Applying SimGNN to the atomistic graph representations of macromolecules is impractical because obtaining a data set with the exact GEDs between atomistic graph representations of macromolecules, which have a large number of atoms within a reasonable time frame is unrealistic. On the other hand, in the coarse-grained graph representations of macromolecules, nodes represent monomers or linkage groups, and one-hot encoding cannot accurately quantify the chemical differences between these nodes nor can it handle the infinite number of possible monomers. Therefore, the direct application of SimGNN to the coarse-grained graph representations of macromolecules reduces the chemical resolution for macromolecule similarity calculation.

Building on the work of SimGNN,[62] this study introduces MacroSimGNN, an extension tailored for macromolecule coarse-grained graph representation pairwise similarity predictions. MacroSimGNN uses Morgan Fingerprints for node embeddings to handle the infinite variety of nodes in the coarse-grained graph representations of macromolecules, accurately quantifying the differences between nodes which represent monomers or linkage groups and allowing extrapolation to unseen nodes. MacroSimGNN aims to overcome the significant drawbacks[47,54−59] of existing approaches by enhancing computational efficiency while preserving high accuracy. Furthermore, MacroSimGNN achieves zero-shot predictions on the new macromolecule data set with high accuracy and performs effective transfer learning to improve predictions using only a small amount of data from the new data set while SimGNN is not able to do these due to the one-hot encoding limitation. MacroSimGNN is then applied along with landmark distance embedding[65,66] for both unsupervised and supervised learning tasks. This work has potential applications in macromolecule search, as well as quantitative design tools for macromolecules.

METHODS

MacroSimGNN

As shown in Figure 1a, in the coarse-grained graph representations of macromolecules, each node is a monomer or linkage group. In real-world scenarios, macromolecules are not formed directly from atoms in a single step; instead, they are synthesized through the polymerization

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

of monomers. Compared to atomistic graphs, coarse-grained graphs clearly distinguish the monomers, providing an accurate reflection of the chemical information. Additionally, the choice of coarse-grained graphs enables the calculation of the GED between the macromolecules in a reasonable time frame, which is not possible if a condensed atomistic graph representation[67,68] is used.[62] The embedding of each node (monomer or linkage group) is achieved via a Morgan fingerprint (radius = 3, nBits = 128, useChirality = True).[47] This choice is the same setting as in Mohapatra et al.[47] Edges represent connections between monomers or linkage groups without chemical specificity or directionality in order to align with the frameworks of MacroSimGNN and SimGNN,[62] which do not include edge-specific information.

As illustrated in Figure 1b, the methodology of MacroSimGNN comprises four stages, mirroring those of SimGNN but with modifications to accommodate macromolecular complexities. Stage 1 inludes graph convolutional networks (GCNs) for node-level embeddings. Nodes are initially embedded using Morgan molecular fingerprints,[47] where $u_{i,n}^0 \in \mathbb{R}^D$ is a 128 dimension vector for node n of graph g_i which has N nodes, differentiating this initial stage from SimGNN[62] which uses one-hot encoding. The graph convolution operation generates the node embeddings for a set of nodes in graph g_i, $U_i \in \mathbb{R}^{N\times D}$, where the n-th row, $u_{i,n} \in \mathbb{R}^D$ is the embedding of node n after graph convolution operation. Stage 2 is graph-level embedding, where an embedding vector for each graph (h_i) is generated by aggregating the input node embeddings (U_i). The node weights are dependent on the similarity matric and are learned and optimized during the training process. Stage 2 is for the neural tensor network used in Stage 3; it is not used for the pairwise node comparison in Stage 3. Stage 3 is for graph−graph interactions including the neural tensor network[62,69] and the pairwise node comparison.[62] The neural tensor network models the relationship between two graph-level embeddings.

$$p(h_i, h_j) = f\left(h_i^T W^{[1:K]} h_j + V\begin{bmatrix} h_i \\ h_j \end{bmatrix} + b\right) \tag{1}$$

where $W^{[1:K]} \in \mathbb{R}^{D\times D\times K}$ is a weight tensor, [] denotes the concatenation operation, $V \in \mathbb{R}^{K\times 2D}$ is a weight vector, $b \in \mathbb{R}^K$ is a bias vector and $f(\cdot)$ is a ReLU activation function. K is a hyperparameter that defines the dimension of the graph-level interaction scores $p(h_i,h_j)$.

However, if only the neural tensor network was used, the node-level information such as the node feature distribution and graph size may be lost by the graph-level embedding.[62] The differences between two graphs lie in small substructures and are usually hard to reflect in graph-level embedding. To overcome this limitation, the pairwise node-level interaction score is obtained by $S = U_i U_j^T$, through matrix multiplication. Next, a normalized histogram feature vector q (hist (S)) is created and concatenated with the graph-level interaction scores $p(h_i,h_j)$. Stage 4 is a fully connected neural network which predicts the similarity score, $\hat{s}(g_i,g_j)$ between graphs g_i and g_j. $\hat{s}(g_i,g_j)$ is compared against the ground-truth similarity score $s(g_i,g_j)$ using the following mean squared error loss function:

$$\text{Loss} = \frac{1}{|\mathcal{D}|}\sum_{(i,j)\in\hat{s}(g_i,g_j)} (\hat{s}(g_i, g_j) - s(g_i, g_j))^2 \tag{2}$$

where $\mathcal{D}$ is the set of training graph pairs.

During training, no explicit symmetry constraint is imposed on the MacroSimGNN framework. Instead, the model learns physical symmetry from the inherently symmetric training data, where the ground truth of $s(g_i,g_j)$ and $s(g_j,g_i)$ are both used for training MacroSimGNN. As a result, the predicted values of $\hat{s}(g_i,g_j)$ and $\hat{s}(g_j,g_i)$ are very close, though slightly different due to the regression nature of the task. For predictions on testing data sets, a symmetry-enforced prediction strategy is implemented to ensure strictly symmetric results and improve prediction accuracy. This symmetry strategy uses the average value $(\hat{s}(g_i,g_j) + \hat{s}(g_j,g_i))/2$ as the final similarity score prediction for the graph pair (g_i,g_j). Detailed discussion and mathematical proof of the benefit of using this postprocessing symmetry strategy compared to not using a postprocessing symmetry strategy are provided in the Supporting Information. Furthermore, an inherently symmetric architecture of MacroSimGNN is developed by transforming the asymmetric neural tensor network into a symmetric one, and a direct comparison of this postprocessing symmetry strategy with an inherently symmetric architecture is also provided in the Supporting Information. It was found that the inherently symmetric architecture performed worse than using a postprocessing strategy. The hyperparameters of MacroSimGNN are tuned by minimizing the mean squared error loss function on the validation data set through a grid search. The details of the optimized hyperparameters are included in the Supporting Information.

In summary, there are two key distinctions between MacroSimGNN and SimGNN.[62] First, MacroSimGNN uses Morgan fingerprints instead of the one-hot node encodings used in SimGNN. This modification enables MacroSimGNN to handle an unlimited variety of possible nodes in the coarse-grained graph representations of macromolecules, to accurately quantify differences between nodes representing monomers, and to extrapolate to unseen nodes through zero-shot predictions and transfer learning. Second, MacroSimGNN explicitly enforces symmetry in similarity prediction for both improved performance and correctness. The adapted framework of MacroSimGNN allows for efficient macromolecule similarity predictions while maintaining the core strengths of the SimGNN approach.

Landmark Distance Embedding

To demonstrate the utility of macromolecule pairwise similarity, it is used as the basis for macromolecule embedding. For comparison, a typical workflow for macromolecule property prediction using AI models involves first obtaining the embedding vectors for the macromolecules, and then inputting these vectors into the models to predict properties. Pairwise similarity or distance is a key component of many common AI algorithms, such as Gaussian processes, k-nearest neighbors, neural networks, and generative models. For example, a Gaussian process takes in the features/embedding vectors and then uses those features/embedding vectors to calculate the pairwise distance matrix using the default euclidean distance function, $d(x_i,x_j)$. This information is then input into the kernel function $k(x_i,x_j)$. For instance, in the case of the radial basis function kernel, the kernel value depends directly and solely on the distance between x_i and x_j. Predictions for a new data point are then made by determining the distances between that new data point and all of the training data. Pairwise distance is also important for neural networks and generative AI. For example, the reconstruction loss of variational autoencoders is a measure of the difference between the input macromolecule and the generated macromolecule.

$$k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{l^2}\right) = f(d(x_i, x_j)) \tag{3}$$

Crafting robust and accurate embedding vectors for each macromolecule is challenging due to the architectural complexity of the macromolecules. As an alternative, precomputing and directly providing the pairwise distance matrix can be a practical solution. In this study, the distances may refer to graph edit distances (GEDs), normalized GEDs (NGEDs), or dissimilarity values calculated as $d = 1 - s$ where s represents similarity. However, this approach has a limitation: directly providing a precomputed distance matrix is only compatible with certain machine learning algorithms. Not all algorithms can accept a distance matrix as input. For example, the random forest algorithm does not use precomputed distance matrices. To address this limitation, an additional method called landmark distance embedding has been developed, which builds upon the idea of using precomputed distances. Landmark distance embedding leverages pairwise distances between entities as embedding vectors, as opposed to crafting embedding vectors for each macromolecule. The approach has previously been used in small molecule property predictions[65,66] and is particularly useful for macromolecules where a simple embedding may not exist due to the architectural complexity. In this work, as illustrated in Figure 2, this method utilizes the pairwise distances of macro-

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

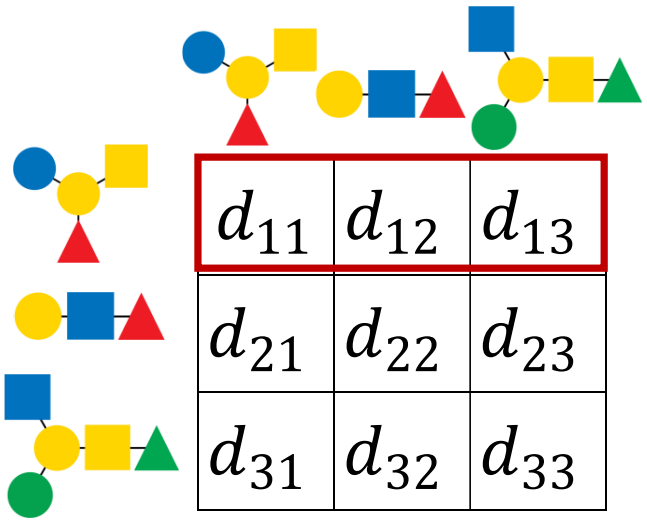


Figure 2. Landmark distance embedding method utilizes the pairwise distances of macromolecules as their embedding vectors.

molecules as their embedding vectors. Detailed discussion of using landmark distance embeddings compared to using the Morgan Fingerprints of the whole macromolecules is provided in the Supporting Information.

While there are certainly many valid and direct ways beyond Morgan Fingerprints to embed macromolecules based on their molecular structural pattern through graph neural networks[53,67,68,70,71] and language models,[28,72−74] previous literature has shown that similarity-based embeddings can be useful for various tasks.[65,66] The focus of this paper is the calculation of macromolecular similarities; this study simply demonstrates immunogenicity prediction with similarity embeddings as a demonstration of the utility of the similarity scores for polymers.

Calculating exact GEDs which are NP-hard, is inefficient and impractical for all pairwise combinations in landmark embedding. Nevertheless, the development of MacroSimGNN has enabled the efficient and accurate computation of pairwise GEDs, NGEDs, and dissimilarity, thus rendering the landmark distance embedding method feasible for macromolecules. Landmark distance embeddings are utilized for unsupervised learning and supervised learning tasks. In this work, specifically, principal component analysis (PCA),[75−78] a linear dimensionality reduction technique, implemented in scikit-learn[79] (*sklearn.decomposition.PCA*) with the number of components being 2, is used for data analysis and visualization of the landmark distance embeddings. Gaussian process classification[31,33,80−83] implemented in scikit-learn[79] (*sklearn.gaussian_process.GaussianProcessClassifier*) with the kernel setting being a combination of constant kernel and radial basis function kernel, is utilized to determine whether a glycan is nonimmunogenic or immunogenic. The results highlight the advantages of macromolecular similarity rather than suggesting similarity is the optimal embedding scheme.

Data Set

Macromolecule Data Set and Data Set Preprocessing. This study utilizes two macromolecular data sets to train MacroSimGNN. The first data set is a glycan data set, originally from GlycoBase[84] and compiled by Mohapatra et al.[47] due to the topological diversity, encompassing both linear and nonlinear configurations, as well as the breadth of monomer chemistries (946 types).[47] This variety makes this data set an ideal test case for the robustness of MacroSimGNN. From the original data set of 19,147 glycans,[47] 400 glycan coarse-grained graph representations are randomly selected for this study. In the following section, *Impact of the Training Data set Size*, this sample size of 10^4 GEDs formed by about 100 glycan graph representations is proved to be sufficient for training MacroSimGNN. Further increasing data size does not provide a noticeable improvement in prediction performance but does require larger memory capacity and longer training time. The distributions of node and edge counts in these 400 graphs are illustrated in Figure 3a,b, respectively. The exact GEDs for all 160,000 pairwise combinations of the 400 selected graphs are calculated by using the A^* algorithm[85] implemented in NetworkX.[86] In the setting of the GED calculation, the cost for each operation of deletion and insertion of nodes and edges is 1; the cost for node substitution is based on the Tanimoto dissimilarity[87] between the two nodes;[47,54] there is no edge substitution. The distribution and matrix of pairwise exact GEDs are shown in Figure 3c,d. These 160,000 pairwise GEDs constitute the macromolecule GED data set.

To preprocess the data for training MacroSimGNN, the ground truth absolute GED(g_1,g_2) is transformed into a similarity score $s(g_1,g_2)$ within the range 0 and 1.[54,55,62] First, the absolute GED is normalized to be NGED

$$\mathrm{NGED}(g_1, g_2) = \frac{\mathrm{GED}(g_1, g_2)}{(N_1 + N_2)/2} \tag{4}$$

where N_i denotes the number of nodes of the graph g_i. NGED(g_1,g_2) is 0 when graph g_1 and g_2 are identical. NGED(g_1,g_2) is symmetric such that NGED(g_1,g_2) = NGED(g_2, g_1). The distribution and matrix of pairwise NGED are shown in Figure 3e,f.

Then an exponential decay function is used to transform the NGED(g_1,g_2) to a similarity score $s(g_1,g_2)$[55,62]

$$s(g_1, g_2) = \exp(-\alpha\cdot\mathrm{NGED}(g_1, g_2)) = \exp\left(-\frac{\alpha\cdot\mathrm{GED}(g_1, g_2)}{(N_1 + N_2)/2}\right) \tag{5}$$

where α is a tunable parameter with the default value being 1. $s(g_1,g_2)$ equals 1 when g_1 and g_2 are identical and approaches 0 as dissimilarity increases. $s(g_1,g_2)$ is also symmetric. The distribution of similarity scores and the heatmap of the pairwise similarity score matrix are illustrated in Figure 3g,h. This transformation ensures a one-to-one mapping between GED and s, while scaling the values to a range between 0 and 1. While the ranges of GED and NGED depend on the specific macromolecule data sets, the range of s is constrained strictly between 0 and 1. Therefore, s served as the target values for training the MacroSimGNN model. GED and NGED are then calculated based from the prediction values of s using eqs 4 and 5.

This work adopts a different data splitting method than the original SimGNN by Bai et al.[62] in order to comprehensively evaluate the prediction ability and generalizability of MacroSimGNN. Based on the distribution of the number of nodes, 200 graphs are randomly selected out of the 400 graphs and reindexed from 1 to 200, with the remaining 200 graphs reindexed from 201 to 400. As shown in Figure 3h, the black region represents the Training data set, which comprises graph pairs from the first 200 graphs. This Training data set is further randomly divided into training (80%) and validation (20%) subsets to reduce overfitting. The red region represents the Testing-1 data set, where one graph in the graph pairs exists in the Training data set. The orange region represents the Testing-2 data set, where neither graph in the graph pair exists in the Training data set. The separation of Testing-1 and Testing-2 data sets aims to comprehensively assess MacroSimGNN's prediction ability and generalizability for similarity between unknown graphs. Equal graph pairs are excluded from all data sets because there are more efficient ways to detect equal graph pairs, and including equal graph pairs in the training hurts the model's performance. The rationale for this exclusion is detailed in the Supporting Information. With 400 equal graph pairs excluded, the actual size of the Training data set is 39,800, and the actual size of the Testing-2 data set is also 39,800. There are no equal graph pairs in the Testing-1 data set; therefore, the size of the Testing-1 data set is 80,000.

The second data set is an antimicrobial peptide (AMP) data set, originally from the database of antimicrobial activity and structure of peptides (DBAASP)[88] and compiled by Mohapatra et al.[47] The purpose of including a second macromolecule data set is to demonstrate the generalizability of MacroSimGNN across different types of macromolecules with completely different chemistry, as well as to showcase its capabilities in zero-shot prediction and transfer learning. From the original data set of 15,778 peptide,[47] 200 AMP coarse-grained graph

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

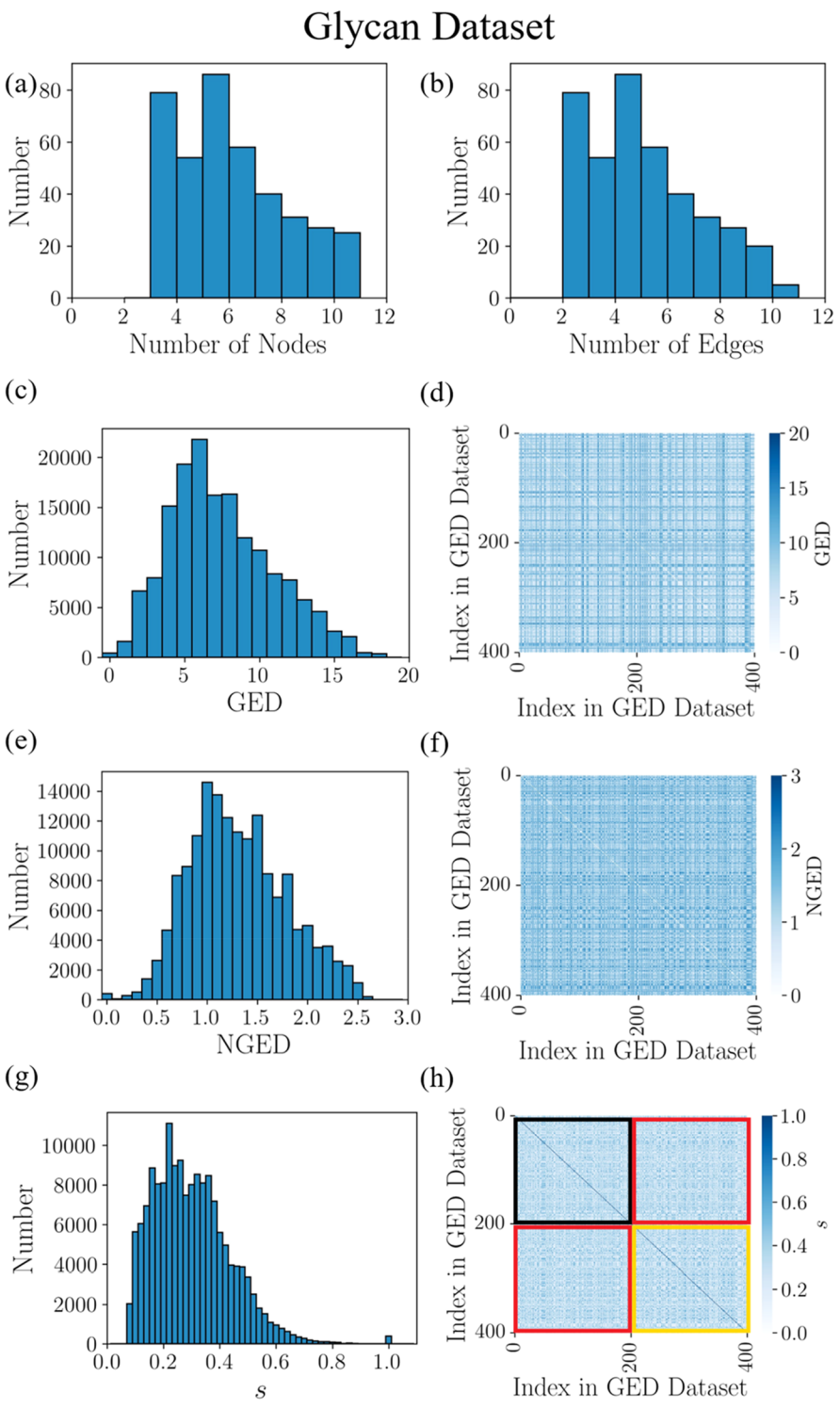


Figure 3. Characteristics of the macromolecular (glycan) data set and derived graph similarity metrics. (a) Distribution of node counts in 400 glycan coarse-grained graph representations. (b) Distribution of edge counts in 400 glycan coarse-grained graph representations. (c) Distribution of pairwise GEDs for 160,000 pairwise comparisons formed by 400 unique macromolecular graphs. (d) Heatmap of the pairwise GED matrix. (e) Distribution of pairwise NGEDs. (f) Heatmap of the pairwise NGED matrix. (g) Distribution of pairwise similarity scores. (h) Heatmap of the pairwise similarity score matrix, as well as the data splitting strategy. The black square represents the Training data set (randomly split 4:1 for training and validation). The red squares represent the Testing-1 data set (pairs with one graph from the Training data set). The orange square represents the Testing-2 data set (pairs with neither graph from the Training data set). This splitting strategy enables a comprehensive evaluation of MacroSimGNN's prediction ability and generalizability for similarity between known and unknown graphs.

representations are randomly selected for this study. The AMP data set is then preprocessed using the same workflow as the glycan data set. The results of the data preprocessing for the AMP data set are shown in Figure 4.

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

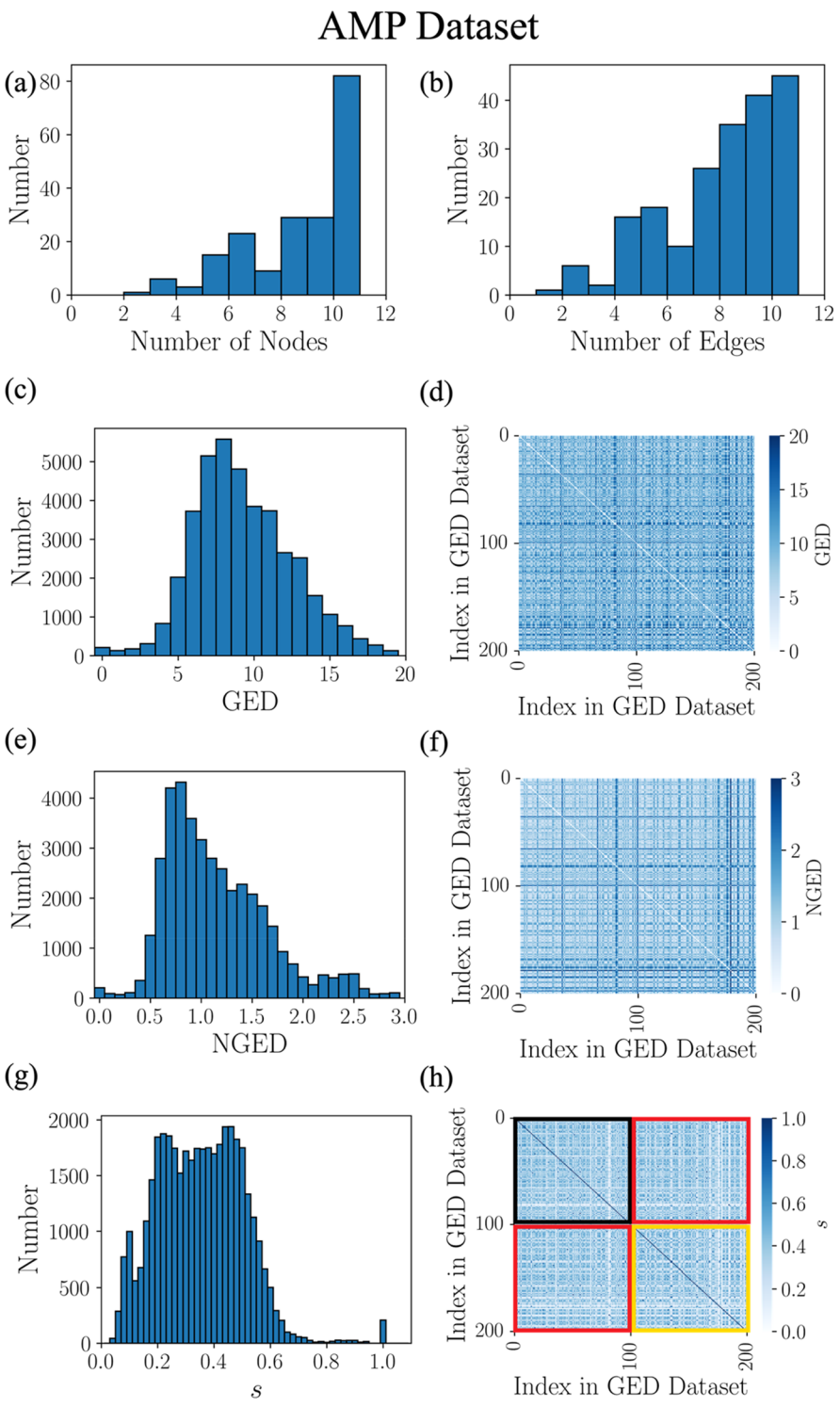


Figure 4. Characteristics of the macromolecular (antimicrobial peptide (AMP)) data set and derived graph similarity metrics. (a) Distribution of node counts in 200 AMP coarse-grained graph representations. (b) Distribution of edge counts in 200 AMP coarse-grained graph representations. (c) Distribution of pairwise GEDs for 40,000 pairwise comparisons formed by 200 unique macromolecular (AMP) graphs. (d) Heatmap of the pairwise GED matrix. (e) Distribution of pairwise NGEDs. (f) Heatmap of the pairwise NGED matrix. (g) Distribution of pairwise similarity scores. (h) Heatmap of the pairwise similarity score matrix, as well as the data splitting strategy. The black square represents the Training data set (randomly split 4:1 for training and validation). The red squares represent the Testing-1 data set (pairs with one graph from the Training data set). The orange square represents the Testing-2 data set (pairs with neither graph from the Training data set). This splitting strategy enables a comprehensive evaluation of MacroSimGNN's prediction ability and generalizability for similarity between known and unknown graphs.

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

RESULTS AND DISCUSSION

Prediction Performance of MacroSimGNN on Glycan Data Set

As can be seen in Figure 5, MacroSimGNN effectively predicts s for both partially known (Testing-1) and entirely unknown

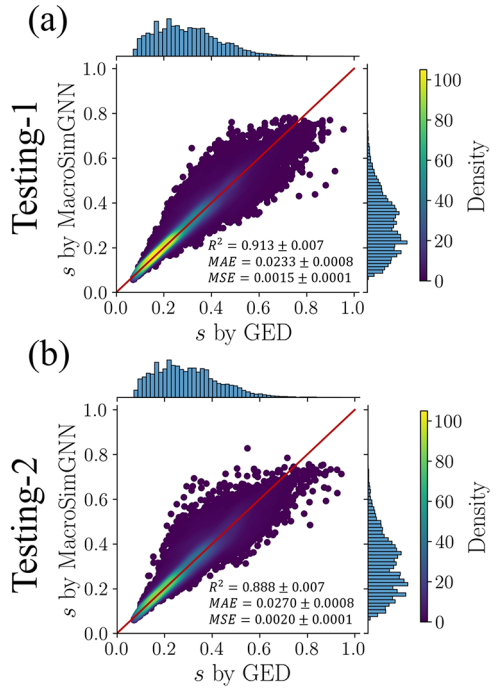


Figure 5. Performances of the MacroSimGNN on Testing-1 and Testing-2 glycan data set for pairwise similarity score s predictions. (a) MacroSimGNN on Testing-1 data set (the R^2 score is 0.913 ± 0.007; the MAE is 0.0233 ± 0.0008; and the MSE is 0.0015 ± 0.0001). (b) MacroSimGNN on Testing-2 data set (the R^2 score is 0.888 ± 0.007; the MAE is 0.0270 ± 0.0008; and the MSE is 0.0020 ± 0.0001).

(Testing-2) graph pairs, showcasing its generalizability and accuracy. Comparing the prediction performance between Testing-1 (where one graph in the pair was seen during training) and Testing-2 (where both graphs were unseen), one finds that MacroSimGNN achieves better predictions when one of the graphs in the pairwise comparison has been encountered during training. This outcome is intuitive. MacroSimGNN uses s by GED as the ground truth. In the Supporting Information, s by GED is directly compared to the graph kernel method and the binary Morgan Fingerprint similarity directly from the whole SMILES strings of the macromolecules. While these are three distinct methods for computing similarity, the distributions for s by GED are broader, and thus GED is better at discriminating between molecules. The predicted s values from MacroSimGNN are then used to compute NGED and GED using eqs 4 and 5, and subsequently compared with the true NGED and GED values, as illustrated in the Supporting Information. Also, the comparison between the prediction performances with and without considering the symmetry are shown in Table S1 in the Supporting Information, indicating that including the symmetry of graph pairs in the predictions makes the prediction strictly symmetric and slightly improves the prediction performance.

The prediction error can be further analyzed. The difference between the predicted s and the true s by GED with respect to the true s by GED is presented in Figure 6a. For the predictions on Testing-1, the Δs values are most densely distributed near zero. When the ground truth $s < 0.5$, the Δs values tend to be greater than zero, whereas for $s > 0.5$, the Δs values tend to be less than zero. Furthermore, predictions corresponding to large ground truth s values exhibit greater errors, likely due to the relatively low frequency of high s values in the data set. Additionally, as shown in Figure 6b, Δs is plotted against $N_1 + N_2$ where $N_1 + N_2$ is the sum of the number of graph nodes in the graph pair (g_1,g_2). The range of Δs remains relatively consistent across different values of $N_1 + N_2$, suggesting that graph size has a small impact on accuracy. The prediction error analysis on Testing-2 has similar features as that on Testing-1, as shown in Figure 6c,d.

Next, the graph pairs of the glycan data set were divided into three structural groups: (1) linear−linear, (2) linear-nonlinear, and (3) nonlinear-nonlinear. MacroSimGNN's prediction performance was then evaluated across these three groups, as shown in Figure 7. The results indicate that the model's performance on linear−linear graph pairs is better than on the other two groups for Testing-1 and significantly so for Testing-2. The entire glycan data set is composed of 167 linear graphs and 233 nonlinear graphs while the training datatset is composed of 91 linear graphs and 109 nonlinear graphs. Therefore, the training data set contains only 20.6% of linear−linear pairs, and the reduced performance of nonlinear graphs is probably not due to the composition of the data set.

Impact of the Training Data Set Size

The impact of the Training data set size on MacroSimGNN model performance is examined by randomly sampling subsets of graphs at various size ratios: 10, 12, 14, 16, 18, 20, 30, 40, 50, 60, 70, 80, and 90% of the 200 graphs which form the full Training data set. For example, at the 10% size ratio, 20 graphs are randomly selected from a total of 200, yielding 380 graph pairs (excluding self-pairs). These 380 graph pairs are randomly divided into a 4:1 ratio for training and validation during the training of MacroSimGNN. This process is repeated five times for each size ratio to ensure statistical robustness. For a fair comparison, the same testing data sets, Testing-1 and Testing-2, are used for the evaluation process. When the training set is reduced, some pairs in Testing-1 data set consist of two graphs that are both unseen during training. Under the reduced-training-size setting, these pairs should in fact be treated as Testing-2, and Testing-1 effectively becomes a mixture of pairs with one seen and one unseen graph and pairs with two unseen graphs. One possible remedy would be to also modify the Testing-1 data set so that it only contains graph pairs in which only one graph appears in the reduced training set. However, this would change the composition of the Testing-1 data set, and the performance comparison between different training data sizes would no longer be consistently based on the same Testing-1 data set but would instead depend on the particular subset chosen. Therefore, the same Testing-1 data set is used for studying the impact of training data size. The most important results are those on the Testing-2 data set, where both graphs in each pair are unseen, reflecting the model's true prediction performance.

For both Testing-1 data set and Testing-2 data set, as the Training data set size increases, the model's performance improves, as evidenced by increases in R^2 (Figure 8) and decreases in MAE (Figure 8) and MSE (Figure S2 in the Supporting Information) for s. Furthermore, the model's

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

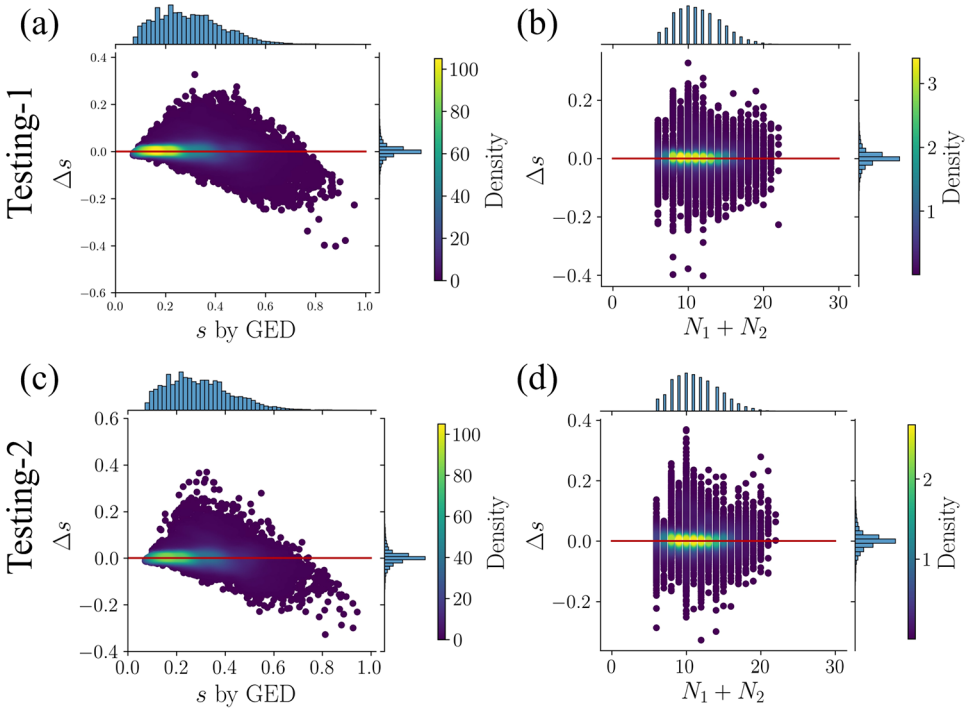


Figure 6. MacroSimGNN prediction error analysis with respect to the ground truth s and $N_1 + N_2$. $N_1 + N_2$ is the sum of the number of graph nodes in the graph pair (g_1,g_2). For Testing-1, (a) Δs vs the s by GED, where $\Delta s = \hat{s} - s$, the difference between the predicted s from MacroSimGNN and the s by GED. (b) Δs vs $N_1 + N_2$. (c, d) is for Testing-2.

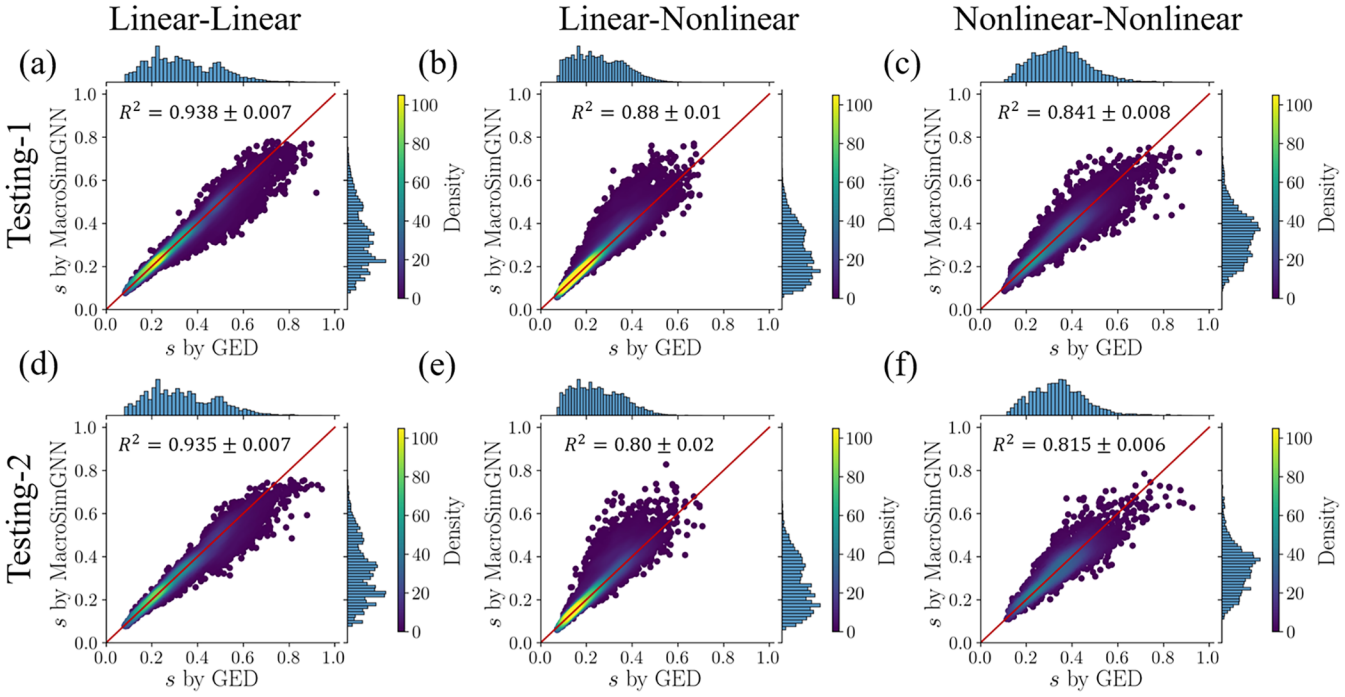


Figure 7. Prediction performance of MacroSimGNN on glycan coarse-grained graph representation pairs across different structural architectures is shown. Panels (a) (Testing-1) and (d) (Testing-2) depict graph pairs in which both graphs are linear. Panels (b, e) show graph pairs where one graph is linear and the other is nonlinear. Panels (c, f) present graph pairs where both graphs are nonlinear.

predictive performance stabilized when the Training data set size reached approximately 10^4. Beyond this point, increases in data set size yielded diminishing returns in performance improvement but increased the cost of memory capacity and computational time. This trend was consistent across both testing data sets and all evaluation metrics.

Prediction Performance Comparison between MacroSimGNN with Morgan Fingerprint and SimGNN with One-Hot Encoding

While one-hot encoding cannot be used on the infinite possibilities of nodes in coarse-grained graph representations of macromolecules, it still serves as a useful benchmark to

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

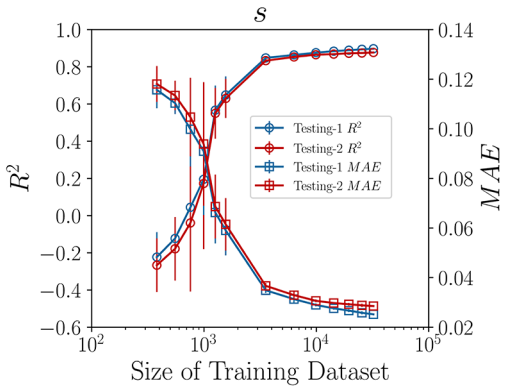


Figure 8. Impact of the Training data set size on the performance of MacroSimGNN predictions on glycan data set. The left *y*-axis shows R^2 of *s* predictions on Testing-1 data set (blue circles) and Testing-2 data set (red circles); the right *y*-axis shows MAE of *s* predictions on Testing-1 data set (blue squares) and Testing-2 data set (red squares). The error bar represents the standard deviation of the five randomly sampled subsets at each size. For both Testing-1 data set and Testing-2 data set, as the Training data set size increases, the model's performance improves, as evidenced by increases in R^2 and decreases in MAE for *s*. Furthermore, when the Training data set size reaches approximately 10^4, increases in data set size yielded diminishing returns in performance improvement of MacroSimGNN.

demonstrate the importance of encoding node chemistry in MacroSimGNN. In this work, for the glycan data set, which contains 946 distinct node types, each node is represented as a 946-dimensional one-hot vector. SimGNN is trained and evaluated using the same workflow as MacroSimGNN on Testing-1 and Testing-2 (including the postprocess symmetry strategy). The performance comparison between MacroSimGNN and SimGNN is shown in Figure 9. For both Testing-1 and Testing-2, MacroSimGNN with Morgan fingerprints demonstrated significantly better performance than SimGNN with one-hot encoding when the available training data was limited. When the training data set size exceeds 10^4, the prediction performance of MacroSimGNN becomes comparable to that of SimGNN, with differences falling within one standard deviation.

Prediction Performance of MacroSimGNN on Antimicrobial Peptide (AMP) Data Set

Another important test for MacroSimGNN is to see if the results extend to another data set with completely different monomer chemistry. For this reason, a second macromolecule data set (AMP data set) is used to demonstrate the generalizability of MacroSimGNN across different types of macromolecules, as well as to showcase its capabilities in zero-shot prediction and transfer learning prediction.

As shown in Figure 10a,c, MacroSimGNN effectively predicts the similarity score (s) for the AMP data set in both partially known (Testing-1) and entirely unknown (Testing-2) graph pairs, demonstrating its generalizability across different types of macromolecules. The R^2 scores for *s* of the AMP data set are higher than the glycan data set. One possible reason is that AMPs are all linear in structure, whereas glycans include both linear and nonlinear structures. MacroSimGNN appears to perform better in predicting pairwise similarity between linear macromolecules. Figure 10b,d further show that, for both Testing-1 and Testing-2, model performance improves with increasing training data set size, as indicated by higher R^2 scores and lower MAE values.

Zero-Shot Prediction of MacroSimGNN

In the prior subsection, MacroSimGNN was retrained on the AMP data set. To determine the generalizability of MacroSimGNN, zero-shot prediction is also examined. In this case, MacroSimGNN is trained on the glycan data set and then tested on the AMP data set. As shown in Figure 11a, the prediction accuracy is good. This result demonstrates that MacroSimGNN is capable of generalizing to completely novel monomer chemistries. The fingerprint-based encoding provides a continuous representation, enabling interpolation to unseen node types—something that one-hot encoding cannot achieve. Figure 11b further shows that the zero-shot prediction performance on the *s* values of the Testing-2 data set of the AMP data set improves as the size of the glycan training data set increases, as evidenced by higher R^2 scores and lower MAE values. When the glycan training data set exceeds approximately 6,320 samples, the zero-shot prediction accuracy on the AMP data set begins to converge. This suggests that a model pretrained on more than 6,320 glycan samples can achieve consistently strong prediction performance on unseen macromolecule types.

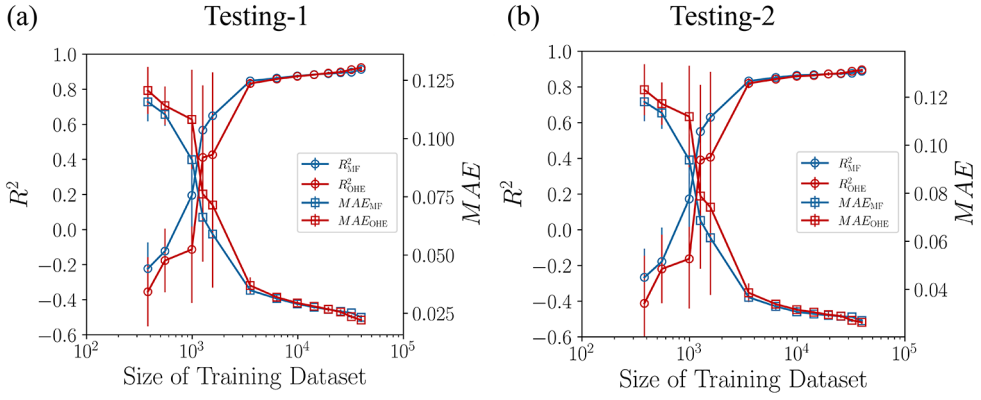


Figure 9. Prediction performance comparison between MacroSimGNN with Morgan Fingerprint (MF) and SimGNN with one-hot encoding (OHE) on Testing-1 (a) and Testing-2 (b) on the glycan data set. For both Testing-1 and Testing-2, when the training data set size is small, MacroSimGNN with Morgan Fingerprints outperforms SimGNN with one-hot encoding. When the training data set size exceeds 10^4, the prediction performance of MacroSimGNN becomes comparable to that of SimGNN, with differences falling within one standard deviation.

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

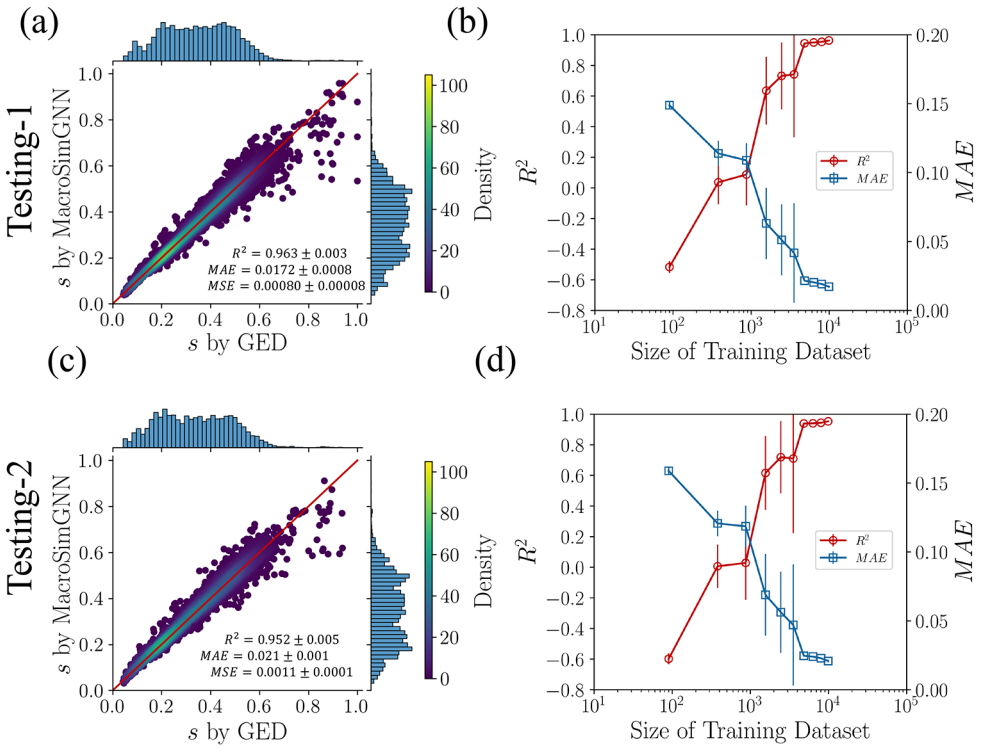


Figure 10. Performances of the MacroSimGNN on Testing-1 (a) and Testing-2 (c) AMP data set for pairwise similarity score *s* predictions. Impact of the training data set size on the performance of MacroSimGNN predictions on AMP data set for Testing-1 (b) and Testing-2 (d).

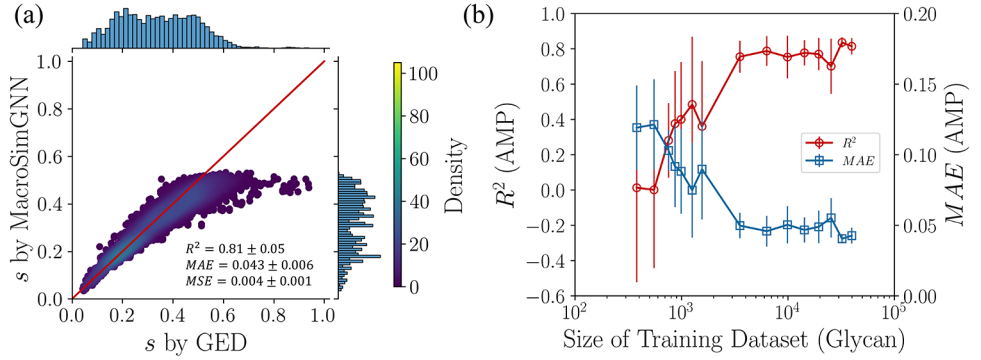


Figure 11. (a) Zero-shot predictions of MacroSimGNN (the pretrained model on the glycan data set) on the pairwise *s* of the Testing-2 of AMP data set. (b) Impact of the Training data set (glycan) size of the pretrained MacroSimGNN model on the zero-hot prediction performance.

Transfer Learning of MacroSimGNN

Next, the transfer learning capability of MacroSimGNN is demonstrated. MacroSimGNN employs Morgan fingerprints for node embedding, which can accommodate a wide variety of monomer molecules. As a result, nodes with different chemical compositions share a consistent embedding dimension, enabling the MacroSimGNN framework to support transfer learning. In the transfer learning (TL) procedure, the MacroSimGNN model is first trained on the glycan data set to obtain a pretrained model. This model is then fine-tuned using data from the AMP data set (Training data set) and then used for prediction on the AMP data set (Testing-2 data set). The prediction performance of TL is compared with direct learning (DL), where the MacroSimGNN model is trained directly on the AMP data set without pretraining. The comparison between TL and DL is presented in Figure 12. The TL approach exhibits significantly better performance than DL, particularly when the available training data is limited.

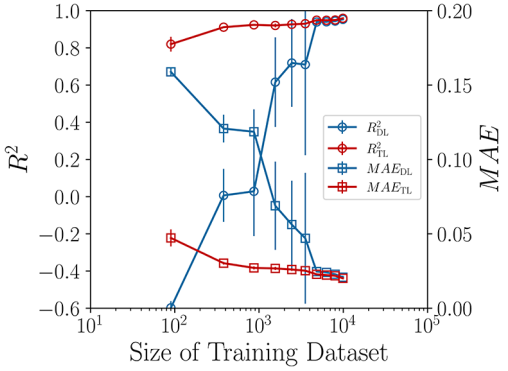


Figure 12. Prediction performance of transfer learning (TL) versus direct learning (DL) of MacroSimGNN on the AMP data set (Testing-2 data set) with varying sizes of training data from the AMP data set (Training data set).

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

Computational Efficiency

As illustrated in Table 1, the computing speed for graph similarity prediction of MacroSimGNN is between the speed of the A^* algorithm and that of the graph kernel. MacroSimGNN demonstrates significantly improved computational efficiency, being over 400 times faster compared to the A^* algorithm. MacroSimGNN and the A^* algorithm calculate the graph pairwise similarity one-by-one, and the details of computing time distributions are shown in Figure S1 in the Supporting Information. The consistently low computation times of MacroSimGNN also suggest improved stability in performance across various graph structures, addressing the high variability often observed with exact methods like the A^* algorithm. All computations were performed on a single core of a MacBook Air M1 CPU to ensure consistent comparison.

Table 1. Computing Efficiency for the A^* Algorithm (Exact GED), MacroSimGNN and the Graph Kernel

method	A^*	MacroSimGNN	graph kernel
average time per one graph pair/second	7.1×10^{-1}	1.6×10^{-3}	5.8×10^{-5}

Landmark Distance Embedding for Unsupervised Learning and Supervised Learning

MacroSimGNN is then applied to develop a landmark distance embedding[65,66] for both unsupervised and supervised learning tasks, using the glycan immunogenicity data set as an example. The glycan immunogenicity data set comprises 470 non-immunogenic and 549 immunogenic glycans. MacroSimGNN is employed to obtain landmark distance embeddings[65,66] for all glycans in this immunogenicity data set. The indices of glycans have been reordered for intuitive visualization: indices 0−469 are nonimmunogenic, and indices 470−1018 are immunogenic, as displayed in the pairwise dissimilarity ($d = 1 - s$) matrix of size 1019×1019 (Figure 13a). Noticeable differences exist between the nonimmunogenic and immunogenic regions.

Each column of the dissimilarity matrix is a landmark distance embedding, which is a 1019-dimensional vector. For unsupervised learning, PCA uses the 1019-dimension landmark distance embedding as the input. The dimensionality reduction results from PCA are depicted in Figure 13b, showing that nonimmunogenic and immunogenic glycans generally occupy distinct locations in the PCA space. Additionally, for supervised learning, the whole glycan immunogenicity data set is divided into training and testing data sets at a ratio 4:1 (Specifically, 815 data samples for training the model and 204 data samples for the held-out test data set). Gaussian process classification using landmark distance embedding as input, predicts immunogenicity with 96.1% accuracy on the held-out test data set, as shown in Figure 13c. The results of using NGED and GED for landmark embedding are illustrated in the Supporting Information, respectively, which are similar to the results of using dissimilarity. For comparison, graph kernel methods have also been used to compute the pairwise similarity and dissimilarity matrices, which are then applied in PCA and Gaussian process classification with 93.6% accuracy on the held-out test data set. The details of the unsupervised learning and supervised learning results built upon the distance matrix calculated by graph kernels[57,58] are demonstrated in the Supporting Information. Comparison indicates that the matrices from MacroSimGNN yield superior distinction in PCA and higher prediction accuracy in Gaussian process classification than those from graph kernel methods.

The next step is to identify the important features of the 1019-dimensional landmark distance embedding for immunogenicity classification. The strong separation of classes in Figure 13b suggests that a small number of features will be sufficient. First, permutation feature importance was performed, where each dimension in the 1019-dimensional embedding was individually set to zero while keeping the remaining 1018 dimensions unchanged. Then the model was retrained. However, no reduction in performance was observed. Randomly selecting 1018 dimensions yielded the same prediction accuracy as that of the full 1019-dimensional embedding. As an alternative, a reverse method was employed and the results are shown in Figure 14. In this approach, each of the 1019 dimensions was individually used to train a Gaussian process classification model, and the resulting prediction accuracy was recorded in Figure 14a, where the indices are the same as the index in the ImG data set in Figure 13a. The prediction accuracy when using only a single dimension ranged from 0.505 to 0.887 as shown in Figure 14b.

Based on these results, the dimensions were ranked from highest to lowest in terms of individual prediction accuracy. Subsequently, the top N-ranked dimensions were incrementally combined to train new models, with the corresponding

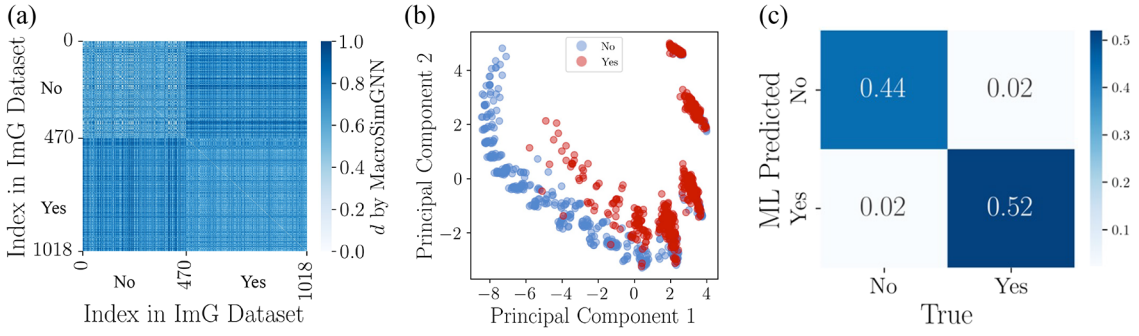


Figure 13. MacroSimGNN is applied to develop landmark distance embeddings for both unsupervised and supervised learning tasks for the glycan immunogenicity (ImG) data set. (a) Pairwise dissimilarity ($d = 1 - s$) matrix where indices 0−469 are nonimmunogenic and 470−1018 are immunogenic. The indices of glycans have been reordered for intuitive visualization. Noticeable differences exist between the nonimmunogenic and immunogenic regions. (b) Dimension reduction results from PCA show that nonimmunogenic (blue) and immunogenic (red) glycans generally occupy distinct locations in the PCA space. (c) Gaussian process classification using landmark distance embedding predicts immunogenicity with 96.1% accuracy on the hold-out test data set.

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

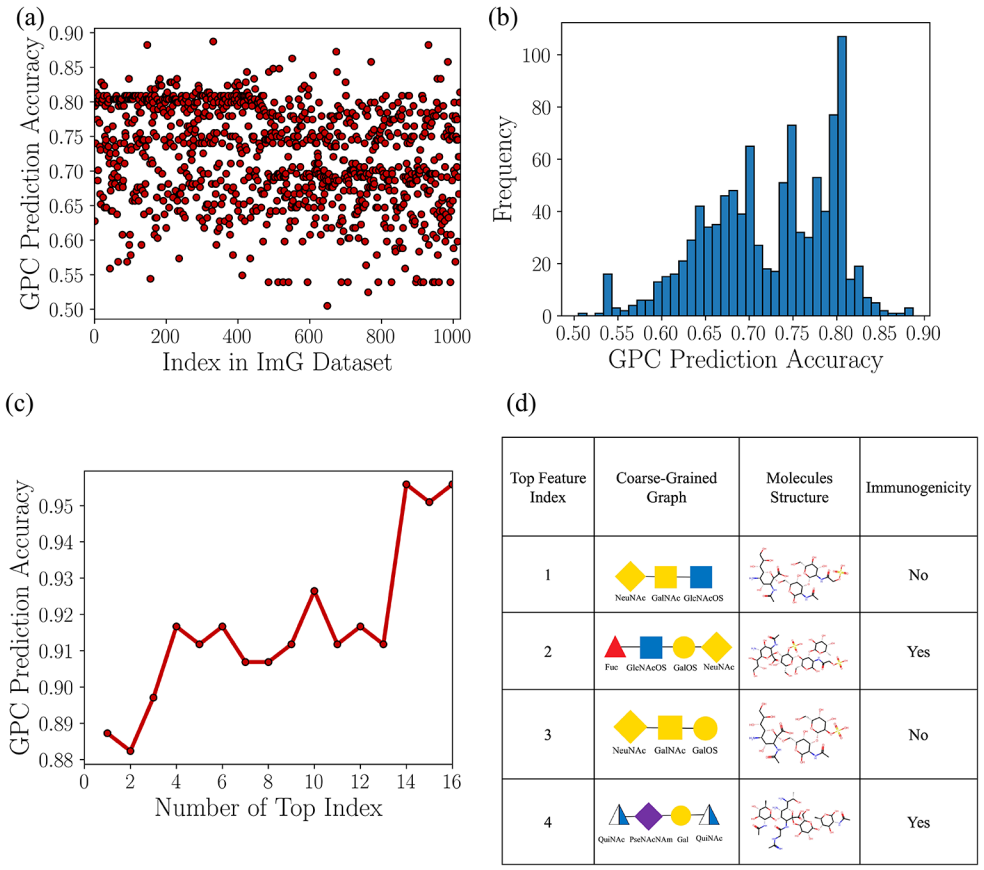


Figure 14. Feature importance analysis of landmark distance embedding. (a) Each of the 1019 dimensions was individually used to train a Gaussian process classification model, and the resulting prediction accuracy was recorded, where the *x*-axis Indices in the ImG Data set are the same as the indices in the ImG Data set in Figure 13a. (b) The distribution of prediction accuracy when using only a single dimension of the landmark distance embedding to train a Gaussian process classification model, ranges from 0.505 to 0.887. (c) The dimensions were ranked from highest to lowest in terms of individual prediction accuracy, and the top N-ranked dimensions were incrementally combined to train new models, with the corresponding performance. (d) The structures of those 4 molecules with the top feature importance.

performance shown in Figure 14c. For example, an *x*-axis value of 10 indicates that the top 10 most predictive dimensions were used for training and prediction. The results demonstrate that using only the top 4 dimensions of the landmark distance embedding yields a prediction accuracy of 91.7%, which is only slightly worse than that of the full model. Since each dimension corresponds to a landmark molecule, we show the structures of these 4 molecules in Figure 14d. Two molecules contain three nodes, while the other two contain four nodes. These molecules differ chemically, composed of different monomer nodes. Two molecules exhibit immunogenic properties and the remaining two are nonimmunogenic.

In addition to interpretability, the method of iteratively adding features can identify more compact models. For example, using 16 dimensions of the landmark distance embedding yields a prediction accuracy of 95.6%, which is comparable to the performance achieved using all 1019 dimensions. This result offers a solution that can significantly reduce the computational cost and resource consumption of generating landmark distance embeddings, while maintaining high predictive performance. This is particularly important in scenarios involving a much larger macromolecule library, where using all molecules as landmarks would result in extremely high-dimensional embeddings and substantial computational overhead.

DISCUSSION OF LIMITATIONS

The coarse-graining process in this study inevitably results in information loss to some extent, and the degree of this loss depends on the complexity of the monomers. Bond chemistry is one of the routes of information loss. In some cases, this bond chemistry may be inferred from the nodes. For example, in this study, although edge types or bond chemistry are not specified, it is still able to infer that the bonds between glycan monomers in the glycan coarse-grained graphs are *O*-glycosidic, while peptide bonds connect amino acid monomers in AMP coarse-grained graphs. As a result, in these scenarios, omitting edge features that represent bonding information does not significantly impact the preservation of essential structural information in the context of this study.

However, for macromolecules composed of complex monomers with asymmetric chemical structures and multiple distinct functional groups, it becomes difficult to determine the exact bonding information solely from the chemical structures of two connected monomers. In such cases, the same pair of monomers may form different types of chemical bonds, resulting in a loss of structural information. These limitations may be developed by changing the way macromolecules are embedded.

Finally, performance improvements of MacroSimGNN could be made depending on the data set by exploring different weightings in the loss function for macromolecules with

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900

different topology. In particular, this could benefit nonlinear macromolecules as previously detailed.

CONCLUSION

This study introduces MacroSimGNN, a graph neural network model designed to accelerate pairwise graph similarity predictions between macromolecules. This model addresses the limitations of previous graph similarity calculation methods, significantly enhancing computational efficiency over 400 times faster than the A^* method while ensuring high accuracy. MacroSimGNN incorporates a physical symmetry strategy during prediction, ensuring strictly symmetric outputs and improving prediction performance. Furthermore, compared to SimGNN, MacroSimGNN demonstrated significantly better performance when training data was insufficient. MacroSimGNN has also been shown to generalize effectively for similarity predictions involving macromolecules with entirely new chemistries, as validated by testing on the AMP data set. Additionally, MacroSimGNN can perform zero-shot predictions with high accuracy and achieve substantial improvements in transfer learning tasks, whereas SimGNN utilizing one-hot encoding is not suitable for these scenarios. Moreover, this study develops landmark distance embeddings derived from MacroSimGNN similarity predictions, achieving promising results in unsupervised and supervised learning tasks, as demonstrated in a case study on glycan immunogenicity. The successful utilization of similarity for embedding underscores the importance of macromolecule similarity in machine learning projects for macromolecules.

The efficient and precise approach of MacroSimGNN has important implications for large-scale analysis and comparison of macromolecular structures, potentially enabling real-time similarity searches in large databases and accelerating the quantitative design of macromolecules.

ASSOCIATED CONTENT

Data Availability Statement

Example scripts and information necessary to run and reproduce all the examples and the corresponding results in this article are posted at the GitHub repository: https://github.com/olsenlabmit/MacroSimGNN.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.macromol.5c03015.

Indirect prediction of NGED and GED by MacroSimGNN; symmetry strategy for graph similarity prediction; equal graph pairs exclusion; MacroSimGNN hyperparameter optimization; distribution of computing times; impact of the training data set size-R^2 and MAE for NGED and GED; impact of the training data set size-MSE; graph kernels for pairwise graph similarity calculation; binary Morgan Fingerprint Tanimoto similarity for graph similarity calculation; landmark distance embedding using NGED and GED from MacroSimGNN; graph kernels for landmark distance embedding; prediction performance comparison between MacroSimGNN (postprocessing symmetric) and MacroSimGNN (architecture symmetric); Morgan Fingerprint vs landmark distance embedding; comparisons of s by GED with the graph kernel method and the binary Morgan Fingerprint similarity method (PDF)

AUTHOR INFORMATION

Corresponding Authors

Bradley D. Olsen − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-7272-7140; Email: bdolsen@mit.edu

Debra J. Audus − *Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, United States;* orcid.org/0000-0002-5937-7721; Email: debra.audus@nist.gov

Authors

Jiale Shi − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, United States;* orcid.org/0000-0002-5447-3925

Runzhong Wang − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States*

Nathan J. Rebello − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-0178-7701

Jiarui Lu − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Department of Computer Science, Wellesley College, Wellesley, Massachusetts 02482, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.macromol.5c03015

Notes

Certain equipment, instruments, software, or materials, commercial or noncommercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose. A version of this manuscript has been previously deposited as a preprint on ChemRxiv (DOI: 10.26434/chemrxiv-2024-8hs2k-v2).
The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was primarily funded by the National Science Foundation Convergence Accelerator award number ITE-2134795. We thank Yunsheng Bai, the first author of SimGNN, for discussions on the details and strategies for training SimGNN.

REFERENCES

(1) Rowan, S. J. 100th Anniversary of Macromolecular Science Viewpoints. *ACS Macro Lett.* **2021**, *10* (4), 466−468.
(2) Sun, H.; Zhong, Z. 100th Anniversary of Macromolecular Science Viewpoint: Biological Stimuli-Sensitive Polymer Prodrugs and Nanoparticles for Tumor-Specific Drug Delivery. *ACS Macro Lett.* **2020**, *9* (9), 1292−1302.
(3) Mohapatra, S.; An, J.; Gómez-Bombarelli, R. In *Graph Attribution Methods Applied to Understanding Immunogenicity in Glycans*, ICML 2021 Workshop on Computational Biology; ICML, 2021.

(4) Varki, A. Biological Roles of Glycans. *Glycobiology* **2017**, *27* (1), 3−49.
(5) Gainza, P.; Wehrle, S.; Van Hall-Beauvais, A.; Marchand, A.; Scheck, A.; Harteveld, Z.; Buckley, S.; Ni, D.; Tan, S.; Sverrisson, F.; Goverde, C.; Turelli, P.; Raclot, C.; Teslenko, A.; Pacesa, M.; Rosset, S.; Georgeon, S.; Marsden, J.; Petruzzella, A.; Liu, K.; Xu, Z.; Chai, Y.; Han, P.; Gao, G. F.; Oricchio, E.; Fierz, B.; Trono, D.; Stahlberg, H.; Bronstein, M.; Correia, B. E. De Novo Design of Protein Interactions with Learned Surface Fingerprints. *Nature* **2023**, *617*, 176−184.
(6) Jones, S.; Thornton, J. M. Principles of Protein-Protein Interactions. *Proc. Natl. Acad. Sci. U.S.A.* **1996**, *93* (1), 13−20.
(7) Lazar, T.; Martínez-Pérez, E.; Quaglia, F.; Hatos, A.; Chemes, L. B.; Iserte, J. A.; Méndez, N. A.; Garrone, N. A.; Saldaño, T. E.; Marchetti, J.; Rueda, A. J. V.; Bernadó, P.; Blackledge, M.; Cordeiro, T. N.; Fagerberg, E.; Forman-Kay, J. D.; Fornasari, M. S.; Gibson, T. J.; Gomes, G. N. W.; Gradinaru, C. C.; Head-Gordon, T.; Jensen, M. R.; Lemke, E. A.; Longhi, S.; Marino-Buslje, C.; Minervini, G.; Mittag, T.; Monzon, A. M.; Pappu, R. V.; Parisi, G.; Ricard-Blum, S.; Ruff, K. M.; Salladini, E.; Skepö, M.; Svergun, D.; Vallet, S. D.; Varadi, M.; Tompa, P.; Tosatto, S. C. E.; Piovesan, D. Ped in 2021: A Major Update of the Protein Ensemble Database for Intrinsically Disordered Proteins. *Nucleic Acids Res.* **2021**, *49* (D1), D404−D411.
(8) Zhao, Y.; Zuo, X.; Li, Q.; Chen, F.; Chen, Y.-R.; Deng, J.; Han, D.; Hao, C.; Huang, F.; Huang, Y.; Ke, G.; Kuang, H.; Li, F.; Li, J.; Li, M.; Li, N.; Lin, Z.; Liu, D.; Liu, J.; Liu, L.; Liu, X.; Lu, C.; Luo, F.; Mao, X.; Sun, J.; Tang, B.; Wang, F.; Wang, J.; Wang, L.; Wang, S.; Wu, L.; Wu, Z.-S.; Xia, F.; Xu, C.; Yang, Y.; Yuan, B.-F.; Yuan, Q.; Zhang, C.; Zhu, Z.; Yang, C.; Zhang, X.-B.; Yang, H.; Tan, W.; Fan, C. Nucleic Acids Analysis. *Sci. China Chem.* **2021**, *64* (2), 171−203.
(9) Opalinska, J. B.; Gewirtz, A. M. Nucleic-Acid Therapeutics: Basic Principles and Recent Applications. *Nat. Rev. Drug Discovery* **2002**, *1* (7), 503−514.
(10) Eschenmoser, A. Chemical Etiology of Nucleic Acid Structure. *Science* **1999**, *284* (5423), 2118−2124.
(11) Provin, A. P.; de Aguiar Dutra, A. R.; Machado, M. M.; Vieira Cubas, A. L. New Materials for Clothing: Rethinking Possibilities through a Sustainability Approach - a Review. *J. Cleaner Prod.* **2021**, *282*, No. 124444.
(12) Geise, G. M.; Lee, H. S.; Miller, D. J.; Freeman, B. D.; McGrath, J. E.; Paul, D. R. Water Purification by Membranes: The Role of Polymer Science. *J. Polym. Sci., Part B: Polym. Phys.* **2010**, *48* (15), 1685−1718.
(13) Guo, Y. H.; Bae, J.; Fang, Z. W.; Li, P. P.; Zhao, F.; Yu, G. H. Hydrogels and Hydrogel-Derived Materials for Energy and Water Sustainability. *Chem. Rev.* **2020**, *120* (15), 7642−7707.
(14) Diao, H.; Yan, F.; Qiu, L.; Lu, J.; Lu, X.; Lin, B.; Li, Q.; Shang, S.; Liu, W.; Liu, J. High Performance Cross-Linked Poly(2-Acrylamido-2-Methylpropanesulfonic Acid)-Based Proton Exchange Membranes for Fuel Cells. *Macromolecules* **2010**, *43* (15), 6398−6405.
(15) Yadav, R.; Tirumali, M.; Wang, X.; Naebe, M.; Kandasubramanian, B. Polymer Composite for Antistatic Application in Aerospace. *Def. Technol.* **2020**, *16* (1), 107−118.
(16) Pendhari, S. S.; Kant, T.; Desai, Y. M. Application of Polymer Composites in Civil Construction: A General Review. *Compos. Struct.* **2008**, *84* (2), 114−124.
(17) Stenzel, M. H. Glycopolymers for Drug Delivery: Opportunities and Challenges. *Macromolecules* **2022**, *55* (12), 4867−4890.
(18) Le, T.; Epa, V. C.; Burden, F. R.; Winkler, D. A. Quantitative Structure-Property Relationship Modeling of Diverse Materials Properties. *Chem. Rev.* **2012**, *112* (5), 2889−2919.
(19) Ma, R. M.; Liu, Z. Y.; Zhang, Q. W.; Liu, Z. Y.; Luo, T. F. Evaluating Polymer Representations Via Quantifying Structure-Property Relationships. *J. Chem. Inf. Model.* **2019**, *59* (7), 3110−3119.
(20) 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; IEEE, 2011; pp 22−29.
(21) Ma, R. M.; Luo, T. F. Pi1m: A Benchmark Database for Polymer Informatics. *J. Chem. Inf. Model.* **2020**, *60* (10), 4684−4690.
(22) Doan Tran, H.; Kim, C.; Chen, L.; Chandrasekaran, A.; Batra, R.; Venkatram, S.; Kamal, D.; Lightstone, J. P.; Gurnani, R.; Shetty, P.; Ramprasad, M.; Laws, J.; Shelton, M.; Ramprasad, R. Machine-Learning Predictions of Polymer Properties with Polymer Genome. *J. Appl. Phys.* **2020**, *128* (17), No. 171104.
(23) Kim, S.; Schroeder, C. M.; Jackson, N. E. Open Macromolecular Genome: Generative Design of Synthetically Accessible Polymers. *ACS Polym. Au.* **2023**, *3*, 318−330.
(24) McGuinness, D.; Brinson, C.; Chen, W.; Daraio, C.; Rudin, C.; Schadler, L.; Cowan, R.; McCusker, J.; Stouffer, S.; Keshan, N. Materialsmine: An Open-Source, User-Friendly Materials Data Resource Guided by Fair Principles 2022; https://tw.rpi.edu/project/materialsmine-open-source-user-friendly-materials-data-resource-guided-fair-principles (accessed September 21, 2023).
(25) Zhao, H.; Li, X.; Zhang, Y.; Schadler, L. S.; Chen, W.; Brinson, L. C. Perspective: Nanomine: A Material Genome Approach for Polymer Nanocomposites Analysis and Design. *APL Mater.* **2016**, *4* (5), No. 053204.
(26) Brinson, L. C.; Deagen, M.; Chen, W.; McCusker, J.; McGuinness, D. L.; Schadler, L. S.; Palmeri, M.; Ghumman, U.; Lin, A.; Hu, B. Polymer Nanocomposite Data: Curation, Frameworks, Access, and Potential for Discovery and Design. *ACS Macro Lett.* **2020**, *9* (8), 1086−1094.
(27) Gurnani, R.; Kamal, D.; Tran, H.; Sahu, H.; Scharm, K.; Ashraf, U.; Ramprasad, R. Polyg2g: A Novel Machine Learning Algorithm Applied to the Generative Design of Polymer Dielectrics. *Chem. Mater.* **2021**, *33* (17), 7008−7016.
(28) Kuenneth, C.; Ramprasad, R. Polybert: A Chemical Language Model to Enable Fully Machine-Driven Ultrafast Polymer Informatics. *Nat. Commun.* **2023**, *14* (1), No. 4099.
(29) Webb, M. A.; Jackson, N. E.; Gil, P. S.; de Pablo, J. J. Targeted Sequence Design within the Coarse-Grained Polymer Genome. *Sci. Adv.* **2020**, *6* (43), No. eabc6216.
(30) Patel, R. A.; Borca, C. H.; Webb, M. A. Featurization Strategies for Polymer Sequence or Composition Design by Machine Learning. *Mol. Syst. Des. Eng.* **2022**, *7* (6), 661−676.
(31) Zhang, Y.; Xu, X. J. Machine Learning Glass Transition Temperature of Polymers. *Heliyon* **2020**, *6*, No. e05055.
(32) Lin, C.; Wang, P.-H.; Hsiao, Y.; Chan, Y.-T.; Engler, A. C.; Pitera, J. W.; Sanders, D. P.; Cheng, J.; Tseng, Y. J. Essential Step toward Mining Big Polymer Data: Polyname2structure, Mapping Polymer Names to Structures. *ACS Appl. Polym. Mater.* **2020**, *2* (8), 3107−3113.
(33) Tao, L.; Varshney, V.; Li, Y. Benchmarking Machine Learning Models for Polymer Informatics: An Example of Glass Transition Temperature. *J. Chem. Inf. Model.* **2021**, *61* (11), 5395−5413.
(34) Tao, L.; Chen, G.; Li, Y. Machine Learning Discovery of High-Temperature Polymers. *Patterns* **2021**, *2* (4), No. 100225.
(35) Chen, L.; Pilania, G.; Batra, R.; Huan, T. D.; Kim, C.; Kuenneth, C.; Ramprasad, R. Polymer Informatics: Current Status and Critical Next Steps. *Mater. Sci. Eng., R* **2021**, *144*, No. 100595.
(36) Statt, A.; Kleeblatt, D. C.; Reinhart, W. F. Unsupervised Learning of Sequence-Specific Aggregation Behavior for a Model Copolymer. *Soft Matter* **2021**, *17* (33), 7697−7707.
(37) Shi, J.; Quevillon, M. J.; Amorim Valença, P. H.; Whitmer, J. K. Predicting Adhesive Free Energies of Polymer−Surface Interactions with Machine Learning. *ACS Appl. Mater. Interfaces* **2022**, *14* (32), 37161−37169.
(38) Gormley, A. J.; Webb, M. A. Machine Learning in Combinatorial Polymer Chemistry. *Nat. Rev. Mater.* **2021**, *6* (8), 642−644.
(39) Tamasi, M. J.; Patel, R. A.; Borca, C. H.; Kosuri, S.; Mugnier, H.; Upadhya, R.; Murthy, N. S.; Webb, M. A.; Gormley, A. J. Machine Learning on a Robotic Platform for the Design of Polymer−Protein Hybrids. *Adv. Mater.* **2022**, *34* (30), No. 2201809.
(40) Patra, T. K. Data-Driven Methods for Accelerating Polymer Design. *ACS Polym. Au* **2022**, *2* (1), 8−26.
(41) Arora, A.; Lin, T. S.; Rebello, N. J.; Av-Ron, S. H. M.; Mochigase, H.; Olsen, B. D. Random Forest Predictor for Diblock Copolymer Phase Behavior. *ACS Macro Lett.* **2021**, *10* (11), 1339−1345.

(42) Wu, Z.; Jayaraman, A. Machine Learning-Enhanced Computational Reverse-Engineering Analysis for Scattering Experiments (Crease) for Analyzing Fibrillar Structures in Polymer Solutions. *Macromolecules* **2022**, *55* (24), 11076−11091.
(43) Heil, C. M.; Patil, A.; Dhinojwala, A.; Jayaraman, A. Computational Reverse-Engineering Analysis for Scattering Experiments (Crease) with Machine Learning Enhancement to Determine Structure of Nanoparticle Mixtures and Solutions. *ACS Cent. Sci.* **2022**, *8* (7), 996−1007.
(44) Yang, J.; Tao, L.; He, J.; McCutcheon, J. R.; Li, Y. Machine Learning Enables Interpretable Discovery of Innovative Polymers for Gas Separation Membranes. *Sci. Adv.* **2022**, *8* (29), 9545−9545.
(45) Smith, T. F.; Waterman, M. S. Identification of Common Molecular Subsequences. *J. Mol. Biol.* **1981**, *147* (1), 195−197.
(46) Eddy, S. R. Where Did the Blosum62 Alignment Score Matrix Come From? *Nat. Biotechnol.* **2004**, *22* (8), 1035−1036.
(47) Mohapatra, S.; An, J.; Gomez-Bombarelli, R. Chemistry-Informed Macromolecule Graph Representation for Similarity Computation, Unsupervised and Supervised Learning. *Mach. Learn.-Sci. Technol.* **2022**, *3*, No. 015028.
(48) Wu, W.; Wang, W.; Li, J. Star Polymers: Advances in Biomedical Applications. *Prog. Polym. Sci.* **2015**, *46*, 55−85.
(49) Altintas, O.; Abbasi, M.; Riazi, K.; Goldmann, A. S.; Dingenouts, N.; Wilhelm, M.; Barner-Kowollik, C. Stability of Star-Shaped Raft Polystyrenes under Mechanical and Thermal Stress. *Polym. Chem.* **2014**, *5* (17), 5009−5019.
(50) Danielsen, S. P. O.; Beech, H. K.; Wang, S.; El-Zaatari, B. M.; Wang, X.; Sapir, L.; Ouchi, T.; Wang, Z.; Johnson, P. N.; Hu, Y.; Lundberg, D. J.; Stoychev, G.; Craig, S. L.; Johnson, J. A.; Kalow, J. A.; Olsen, B. D.; Rubinstein, M. Molecular Characterization of Polymer Networks. *Chem. Rev.* **2021**, *121* (8), 5042−5092.
(51) Wu, T.; Guo, Z.; Cheng, J. Atomic Protein Structure Refinement Using All-Atom Graph Representations and Se(3)-Equivariant Graph Transformer. *Bioinformatics* **2023**, *39* (5), No. btad298.
(52) Wang, Y.; Kalscheur, J.; Ebikade, E.; Li, Q.; Vlachos, D. G. Ligningraphs: Lignin Structure Determination with Multiscale Graph Modeling. *J. Cheminf.* **2022**, *14* (1), No. 43.
(53) Wilson, A. N.; St John, P. C.; Marin, D. H.; Hoyt, C. B.; Rognerud, E. G.; Nimlos, M. R.; Cywar, R. M.; Rorrer, N. A.; Shebek, K. M.; Broadbelt, L. J.; Beckham, G. T.; Crowley, M. F. Polyid: Artificial Intelligence for Discovering Performance-Advantaged and Sustainable Polymers. *Macromolecules* **2023**, *56*, 8547−8557.
(54) Shi, J.; Walsh, D.; Zou, W.; Rebello, N.; Deagen, M.; Fransen, K.; Gao, X.; Olsen, B.; Audus, D. Calculating Pairwise Similarity of Polymer Ensembles Via Earth Mover's Distance. *ACS Polym. Au* **2024**, *4*, 66−76.
(55) Shi, J.; Rebello, N. J.; Walsh, D.; Zou, W.; Deagen, M. E.; Leão, B. S.; Audus, D. J.; Olsen, B. D. Quantifying Pairwise Similarity for Complex Polymers. *Macromolecules* **2023**, *56* (18), 7344−7357.
(56) Wu, Z. H.; Pan, S. R.; Chen, F. W.; Long, G. D.; Zhang, C. Q.; Yu, P. S. A Comprehensive Survey on Graph Neural Networks. *Ieee TranS. Neural Networks Learn. Syst.* **2021**, *32* (1), 4−24.
(57) Siglidis, G.; Nikolentzos, G.; Limnios, S.; Giatsidis, C.; Skianis, K.; Vazirgiannis, M. Grakel: A Graph Kernel Library in Python. *J. Mach. Learn. Res.* **2020**, *21* (54), 1−5.
(58) Neumann, M.; Garnett, R.; Bauckhage, C.; Kersting, K. Propagation Kernels: Efficient Graph Kernels from Propagated Information. *Mach. Learn.* **2016**, *102* (2), 209−245.
(59) Kriege, N. M.; Johansson, F. D.; Morris, C. A Survey on Graph Kernels. *Appl. Network Sci.* **2020**, *5*, No. 6.
(60) Sanfeliu, A.; Fu, K.-S. A Distance Measure between Attributed Relational Graphs for Pattern Recognition. *IEEE Trans. Syst., Man, and Cybernetics* **1983**, No. 3, 353−362.
(61) Blumenthal, D. B.; Gamper, J. On the Exact Computation of the Graph Edit Distance. *Pattern Recog. Lett.* **2020**, *134*, 46−57.
(62) Bai, Y. S.; Ding, H.; Bian, S.; Chen, T.; Sun, Y. Z.; Wang, W. *Simgnn: A Neural Network Approach to Fast Graph Similarity Computation*, 12th ACM International Conference on Web Search and Data Mining (WSDM), Melbourne, AUSTRALIA, Feb 11−15, 2019; Assoc Computing Machinery: NEW YORK, 2019; Vol. *2019*, pp 384−392. DOI: 10.1145/3289600.3290967.
(63) Wang, R.; Yan, J.; Yang, X. Combinatorial Learning of Robust Deep Graph Matching: An Embedding Based Approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45* (6), 6984−7000.
(64) Wang, R.; Yan, J.; Yang, X. Neural Graph Matching Network: Learning Lawler's Quadratic Assignment Problem with Extension to Hypergraph and Multiple-Graph Matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44* (9), 5261−5279.
(65) Dong, J.; Varbanov, M.; Philippot, S.; Vreken, F.; Zeng, W.-b.; Blay, V. Ligand-Based Discovery of Coronavirus Main Protease Inhibitors Using Macaw Molecular Embeddings. *J. Enzyme Inhib. Med. Chem.* **2023**, *38* (1), 24−35.
(66) Blay, V.; Radivojevic, T.; Allen, J. E.; Hudson, C. M.; Garcia Martin, H. Macaw: An Accessible Tool for Molecular Embedding and Inverse Molecular Design. *J. Chem. Inf. Model.* **2022**, *62* (15), 3551−3564.
(67) Aldeghi, M.; Coley, C. W. A Graph Representation of Molecular Ensembles for Polymer Property Prediction. *Chem. Sci.* **2022**, *13* (35), 10486−10498.
(68) Antoniuk, E. R.; Li, P.; Kailkhura, B.; Hiszpanski, A. M. Representing Polymers as Periodic Graphs with Learned Descriptors for Accurate Polymer Property Predictions. *J. Chem. Inf. Model.* **2022**, *62* (22), 5435−5445.
(69) Socher, R.; Chen, D.; Manning, C. D.; Ng, A. Y. In *Reasoning with Neural Tensor Networks for Knowledge Base Completion*, In Proceedings of the 26th International Conference on Neural Information Processing Systems; NIPS, 2013.
(70) Zhang, X.; Sheng, Y.; Liu, X.; Yang, J.; Goddard Iii, W. A.; Ye, C.; Zhang, W. Polymer-Unit Graph: Advancing Interpretability in Graph Neural Network Machine Learning for Organic Polymer Semiconductor Materials. *J. Chem. Theory Comput.* **2024**, *20* (7), 2908−2920.
(71) Queen, O.; McCarver, G. A.; Thatigotla, S.; Abolins, B. P.; Brown, C. L.; Maroulas, V.; Vogiatzis, K. D. Polymer Graph Neural Networks for Multitask Property Learning. *npj Comput. Mater.* **2023**, *9* (1), No. 90.
(72) Zhang, T.; Yang, D.-B. Multimodal Machine Learning with Large Language Embedding Model for Polymer Property Prediction. *Chem. Mater.* **2025**, *37* (18), 7002−7013.
(73) Agarwal, S.; Mahmood, A.; Ramprasad, R. Polymer Solubility Prediction Using Large Language Models. *ACS Mater. Lett.* **2025**, *7* (6), 2017−2023.
(74) Gupta, S.; Mahmood, A.; Shukla, S.; Ramprasad, R. Benchmarking Large Language Models for Polymer Property Predictions. **2025**. DOI: 10.1002/marc.202500388.
(75) Greenacre, M.; Groenen, P. J. F.; Hastie, T.; D'Enza, A. I.; Markos, A.; Tuzhilina, E. Principal Component Analysis. *Nat. Rev. Methods Prim.* **2022**, *2* (1), No. 100.
(76) Sidou, L. s. F.; Borges, E. M. Teaching Principal Component Analysis Using a Free and Open Source Software Program and Exercises Applying Pca to Real-World Examples. *J. Chem. Educ.* **2020**, *97* (6), 1666−1676.
(77) Héberger, K.; Milczewska, K.; Voelkel, A. Principal Component Analysis of Polymer−Solvent and Filler−Solvent Interactions by Inverse Gas Chromatography. *Colloids Surf., A* **2005**, *260* (1), 29−37.
(78) Banerjee, A.; Hsu, H.-P.; Kremer, K.; Kukharenko, O. Data-Driven Identification and Analysis of the Glass Transition in Polymer Melts. *ACS Macro Lett.* **2023**, *12* (6), 679−684.
(79) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-Learn: Machine Learning in Python *J. Mach. Learn. Res.*, *12* 2825 2830.
(80) Deringer, V. L.; Bartók, A. P.; Bernstein, N.; Wilkins, D. M.; Ceriotti, M.; Csányi, G. Gaussian Process Regression for Materials and Molecules. *Chem. Rev.* **2021**, *121* (16), 10073−10141.
(81) Frazier, P. I.; Wang, J. Bayesian Optimization for Materials Design. In *Information Science for Materials Discovery and Design*;

Lookman, T.; Alexander, F. J.; Rajan, K., Eds.; Springer International Publishing, 2016; pp 45−75.
(82) Chen, Z.; Li, D.; Liu, J.; Gao, K. Application of Gaussian Processes and Transfer Learning to Prediction and Analysis of Polymer Properties. *Comput. Mater. Sci.* **2023**, *216*, No. 111859.
(83) Obrezanova, O.; Segall, M. D. Gaussian Processes for Classification: Qsar Modeling of Admet and Target Activity. *J. Chem. Inf. Model.* **2010**, *50* (6), 1053−1061.
(84) Bojar, D.; Powers, R. K.; Camacho, D. M.; Collins, J. J. Deep-Learning Resources for Studying Glycan-Mediated Host-Microbe Interactions. *Cell Host Microbe* **2021**, *29* (1), 132−144.e133.
(85) Abu-Aisheh, Z.; Raveaux, R.; Ramel, J. Y.; Martineau, P. In *An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems*, ICPRAM 2015—4th International Conference on Pattern Recognition Applications and Methods, Proceedings; ICPRAM, 2015; pp 271−278.
(86) Networkx 2024; https://networkx.org/ (accessed December 10, 2023).
(87) Bajusz, D.; Rácz, A.; Héberger, K. Why Is Tanimoto Index an Appropriate Choice for Fingerprint-Based Similarity Calculations? *J. Cheminf.* **2015**, *7* (1), 20−20.
(88) Pirtskhalava, M.; Gabrielian, A.; Cruz, P.; Griggs, H. L.; Squires, R. B.; Hurt, D. E.; Grigolava, M.; Chubinidze, M.; Gogoladze, G.; Vishnepolsky, B.; Alekseev, V.; Rosenthal, A.; Tartakovsky, M. Dbaasp V.2: An Enhanced Database of Structure and Antimicrobial/Cytotoxic Activity of Natural and Synthetic Peptides. *Nucleic Acids Res.* **2016**, *44* (D1), D1104−D1112.

https://doi.org/10.1021/acs.macromol.5c03015
Macromolecules 2026, 59, 1885−1900