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

Jiale Shi[1,2], Runzhong Wang[1], Nathan J. Rebello[1], Jiarui Lu[1,3], Bradley D. Olsen[1‡], Debra J. Audus[2‡]

1. Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States
2. Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, United States
3. Department of Computer Science, Wellesley College, Wellesley, Massachusetts 02482, United States

‡Correspondence: email: bdolsen@mit.edu and debra.audus@nist.gov

Abstract

Efficient and accurate calculation 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 enhances machine learning techniques, including clustering, classification, and regression for predicting properties and discovering new macromolecular materials.[19, 27-44]

While sequence matching algorithms[45, 46] can be used for similarity calculations in simple linear macromolecules, many complex macromolecules have non-linear 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 high 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 production operation can lose small but important structural differences between graphs. Therefore, graph kernel methods offer improved efficiency but often suffer from reduced 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 dataset with the exact GEDs between atomistic graph representations of macromolecules, which have a large number of atoms within a reasonable timeframe 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. 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 calculations. MacroSimGNN uses Morgan Fingerprints for node embeddings to accurately quantify the differences between nodes which represent monomers or linkage groups. MacroSimGNN aims to overcome the significant drawbacks[47, 54-59] of existing approaches by enhancing computational efficiency while preserving high accuracy. 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; the Morgan fingerprint (radius = 3, nBits = 128, useChirality=True)[47] of the monomer or linkage group is the embedding of the node, which is the same setting 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 3 is graph-graph interactions including the neural tensor network[62, 67] and the pairwise node comparison.[62] The neural tensor network models the relationship between two graph-level embeddings.

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

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 controlling the number of interaction (similarity) scores produced by the model for each graph embedding pair.

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 $q(\text{hist}(S))$ 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:

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

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 datasets, a symmetry-enforced prediction strategy is implemented to ensure strictly symmetric results and improve prediction accuracy. This symmetry strategy uses the average value $\left(\hat{s}(g_i, g_j) + \hat{s}(g_j, g_i)\right)/2$ as the final similarity score prediction for the graph pair (g_i, g_j). Detailed discussion and mathematical proof of this symmetry strategy are provided in the Supporting Information.

Stages 2 to 4, with the exception of the symmetry strategy, are consistent with the methodology described by Bai et al.[62] This adapted framework allows for efficient macromolecule similarity calculations while maintaining the core strengths of the SimGNN approach. By using coarse-grained representations and Morgan fingerprints, MacroSimGNN can handle the complexity of macromolecules without sacrificing computational efficiency or accuracy. The hyperparameters of MacroSimGNN are tuned by minimizing the mean squared error loss function on the validation dataset through a grid search. The details of the optimized hyperparameters are included in the Supporting Information.

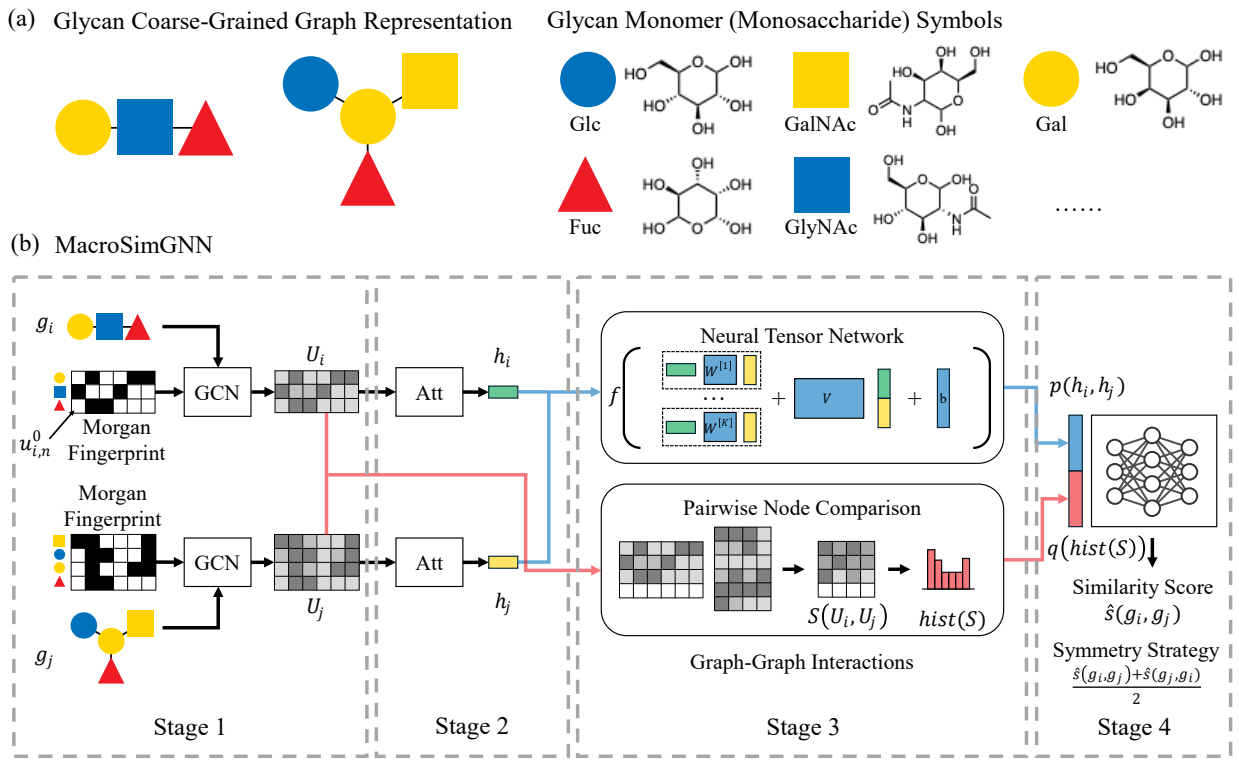


Figure 1: (a) Coarse-grained graph representations of glycans where the nodes represents glycan monomers (monosaccharide). (b) Schematic representation of the MacroSimGNN methodology. The four-stage process of MacroSimGNN, adapted from SimGNN.[62] Stage 1 includes graph convolutional networks (GCNs) 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 3 is 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 calculations while maintaining computational efficiency.

Landmark Distance Embedding

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 macromolecules as their embedding vectors. In this study, these distances may be GEDs, normalized GEDs (NGEDs), or

dissimilarity ($d = 1 - s$). Calculating exact GEDs which is 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)[68-71], a linear dimensionality reduction technique, implemented in scikit-learn[72] (*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, 73-76] implemented in scikit-learn[72] (*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 non-immunogenic or immunogenic.

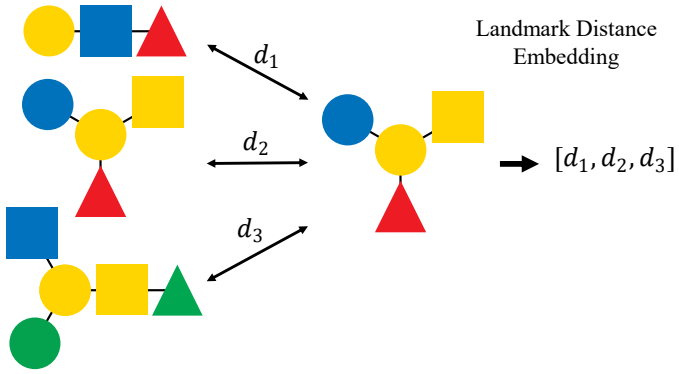


Figure 2: The landmark distance embedding method utilizes the pairwise distances of macromolecules as their embedding vectors.

Dataset

Macromolecule GED Dataset and Dataset Preprocessing

This study utilizes a glycan dataset, originally from GlycoBase[77] 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 dataset an ideal test case for the robustness of MacroSimGNN. From the original dataset 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 Dataset Size*, this sample size of 10^4 GEDs formed by about 100 glycan 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 requires larger memory capacity and longer training time. The distributions of node and edge counts in these 400 graphs are illustrated in Figures 3a and 3b, respectively. The exact GEDs for all 160,000 pairwise combinations of the 400 selected graphs are calculated by using the A*

algorithm[78] implemented in NetworkX.[79] 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[80] between the two nodes;[47, 54] there is no edge substitution. The distribution and matrix of pairwise exact GEDs are shown in Figures 3c and 3d. These 160,000 pairwise GEDs constitute the macromolecule GED dataset.

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,

$$NGED(g_1, g_2) = \frac{GED(g_1, g_2)}{(N_1 + N_2)/2} \quad (1)$$

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 Figures 3e and 3f.

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]

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

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 Figures 3g and 3h. This transformation ensures a one-to-one mapping between GED and s, while scaling the values to a range between 0 and 1.

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 dataset, which comprises graph pairs from the first 200 graphs. This Training dataset is further randomly divided into training (80 %) and validation (20 %) subsets to reduce overfitting. The red region represents the Testing-1 dataset, where one graph in the graph pairs exists in the Training dataset. The orange region represents the Testing-2 dataset, where neither graph in the graph pair exists in the Training dataset. The separation of Testing-1 and Testing-2 datasets aims to comprehensively assess MacroSimGNN's prediction ability and generalizability for similarity between unknown graphs. Equal graph pairs are excluded from all datasets 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 dataset is 39,800, and the actual size of the Testing-2 dataset is also 39,800. There are no equal graph pairs in the Testing-1 dataset; therefore, the size of the Testing-1 dataset is 80,000.

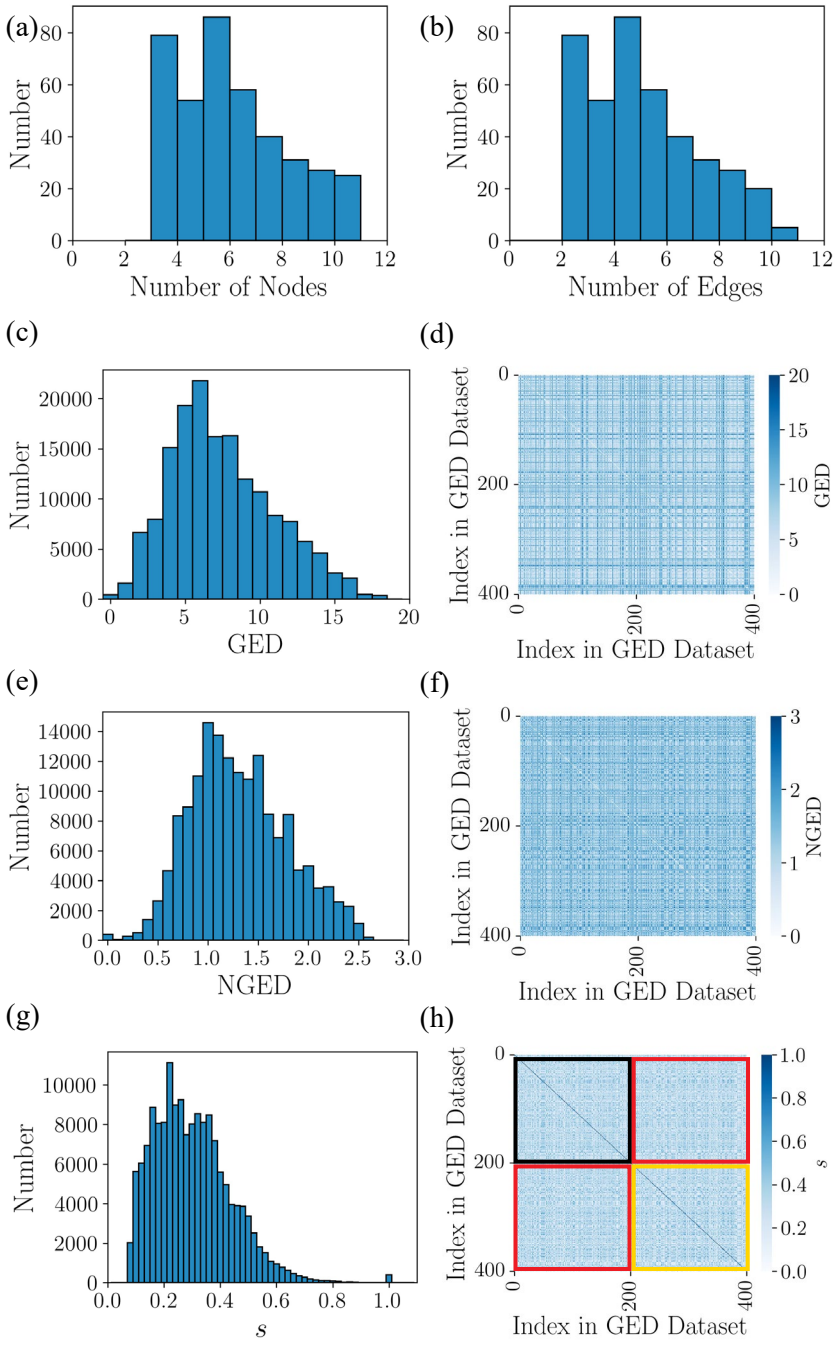


Figure 3: Characteristics of the macromolecular (glycan) dataset and derived graph similarity metrics. (a) Distribution of node counts in 400 glycan coarse-grained graph representations. (b)

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 presents the Training dataset (randomly split 4:1 for training and validation). The red squares represent the Testing-1 dataset (pairs with one graph from the Training dataset). The orange square represents the Testing-2 dataset (pairs with neither graph from the Training dataset). This splitting strategy enables a comprehensive evaluation of MacroSimGNN's prediction ability and generalizability for similarity between known and unknown graphs.

Results and Discussions

Prediction Performance of MacroSimGNN

Figures 4a and c illustrate the predictive accuracies of the MacroSimGNN. For a benchmark comparision, the graph kernel method is chosen. The details about the setting of the graph kernel method and the hyperparameter optimization are provided in the Supporting Information. As can be seen in Figure 4, MacroSimGNN effectively predicts s for both partially known (Testing-1) and entirely unknown (Testing-2) graph pairs, showcasing its generalizability and accuracy. Figure 5 and Figure 6 also demonstrates the higher accuracy of MacroSimGNN in predicting NGED and GED compared to the graph kernel method.

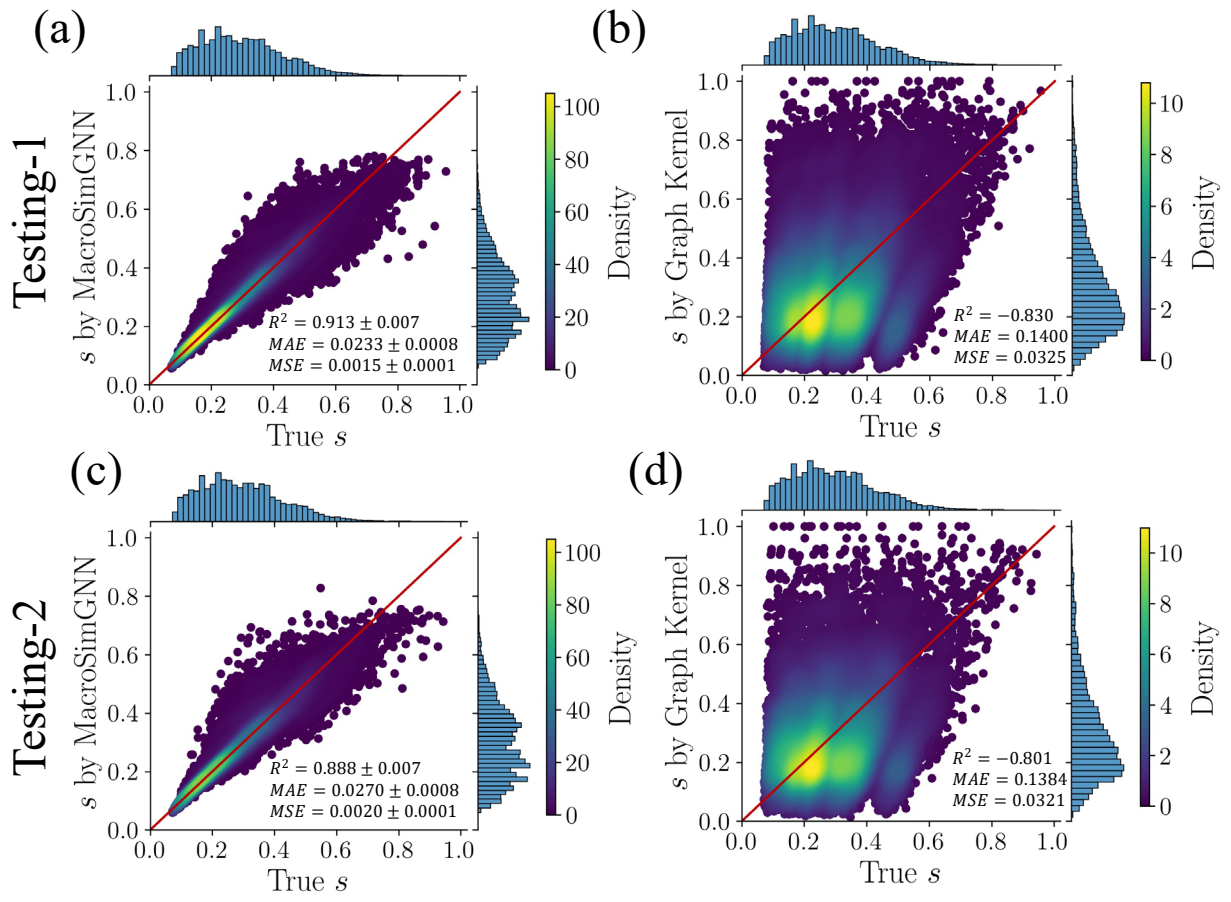


Figure 4: The performances of the MacroSimGNN vs the graph kernel on Testing-1 and Testing-2 dataset for pairwise similarity score s predictions. (a) MacroSimGNN on Testing-1 dataset (the R^2 score is 0.913 ± 0.007; the MAE is 0.0233 ± 0.0008; and the MSE is 0.0015 ± 0.0001). (b) Graph kernel on Testing-1 dataset (the R^2 score is -0.830; the MAE is 0.1400; and the MSE is 0.0325). (c) MacroSimGNN on Testing-2 dataset (the R^2 score is 0.888 ± 0.007; the MAE is 0.0270 ± 0.0008; and the MSE is 0.0020 ± 0.0001). (d) Graph kernel on Testing-2 dataset (the R^2 score is -0.801; the MAE is 0.1384; and the MSE is 0.0321).

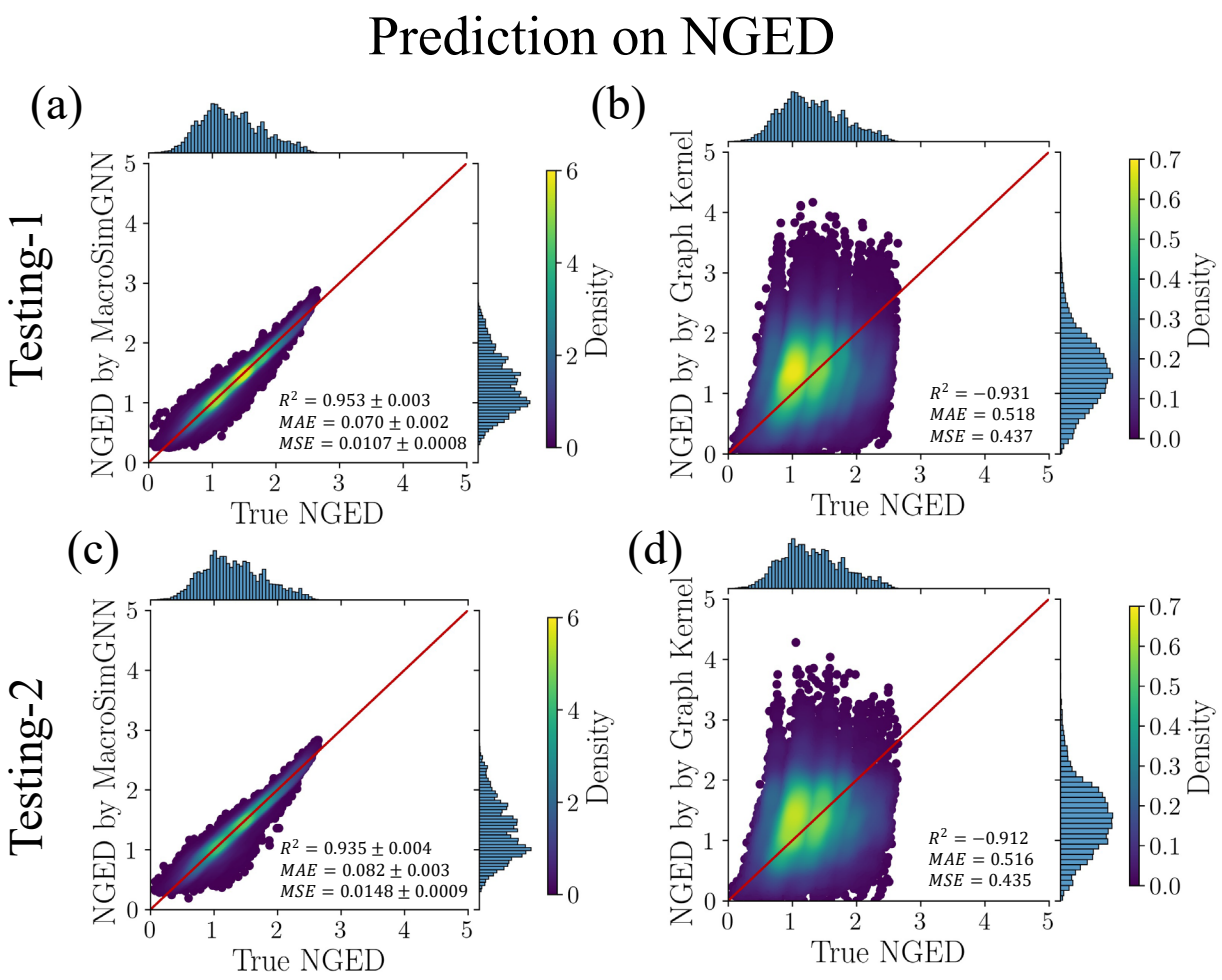


Figure 5: The performances of the MacroSimGNN with the symmetry strategy vs the graph kernel on Testing-1 and Testing-2 dataset for pairwise NGED predictions. (a) MacroSimGNN on Testing-1 dataset (the R^2 score is 0.953 ± 0.003; the MAE is 0.070 ± 0.002; and the MSE is 0.0107 ± 0.0008). (b) Graph kernel on Testing-1 dataset (the R^2 score is -0.931; the MAE is 0.518; and the MSE is 0.437). (c) MacroSimGNN on Testing-2 dataset (the R^2 score is 0.935 ± 0.004; the MAE is 0.082 ± 0.003; and the MSE is 0.0148 ± 0.0009). (d) Graph kernel on Testing-2 dataset (the R^2 score is -0.912; the MAE is 0.516; and the MSE is 0.435).

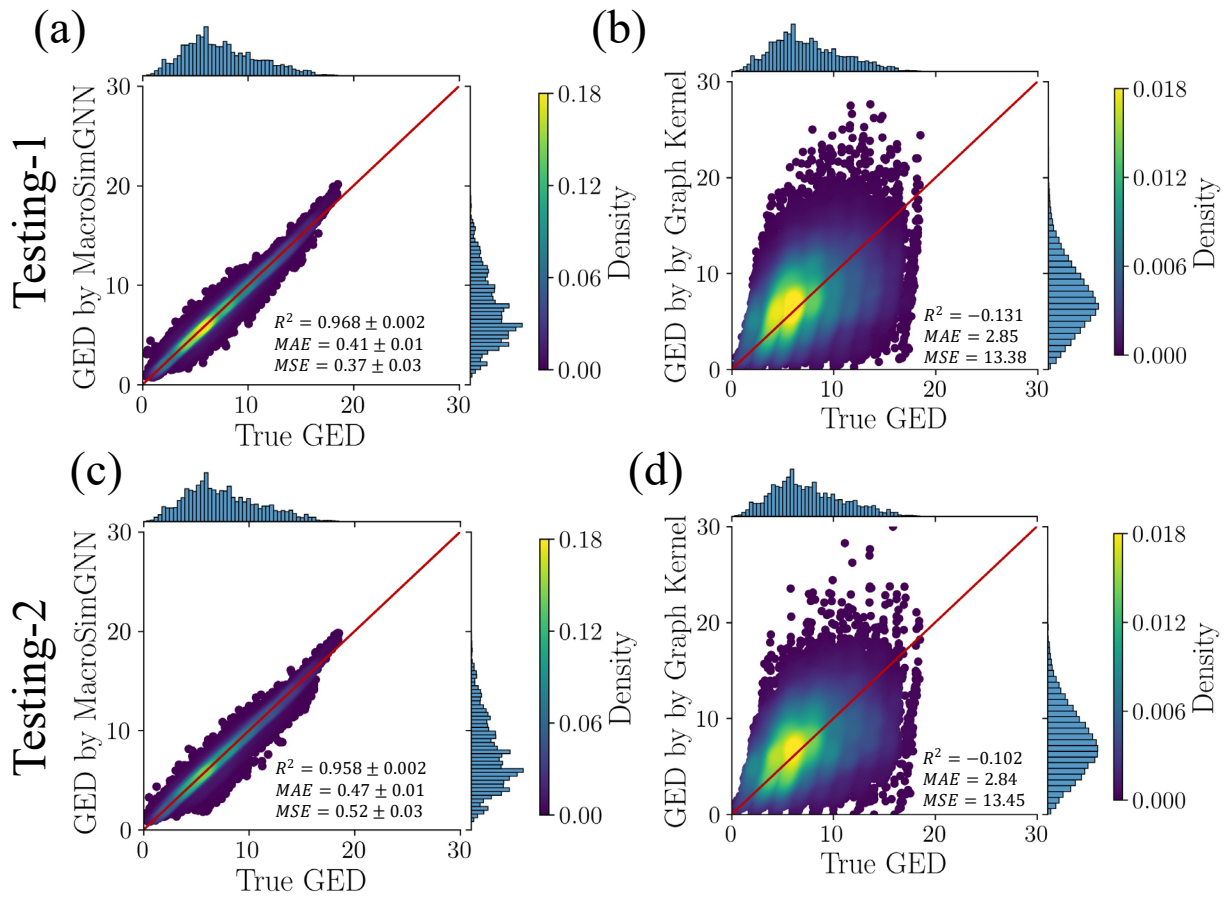


Figure 6: The performances of the MacroSimGNN with the symmetry strategy vs the graph kernel on Testing-1 and Testing-2 dataset for pairwise GED predictions. (a) MacroSimGNN on Testing-1 dataset (the R^2 score is 0.968 ± 0.002; the MAE is 0.41 ± 0.01; and the MSE is 0.37 ± 0.03). (b) Graph kernel on Testing-1 dataset (the R^2 score is -0.131; the MAE is 2.85; and the MSE is 13.38). (c) MacroSimGNN on Testing-2 dataset (the R^2 score is 0.958 ± 0.002; the MAE is 0.47 ± 0.01; and the MSE is 0.52 ± 0.03). (d) Graph kernel on Testing-2 dataset (the R^2 score is -0.102; the MAE is 2.84; and the MSE is 13.45).

As shown in Table 1, evaluation metrics, including coefficient of determination (R^2), mean absolute error (MAE) and mean squared error (MSE), are used to quantify the accuracy of the model's prediction on GED, NGED and s. In line with Figure 4, MacroSimGNN significantly outperforms the Graph Kernel method. 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. 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.

Table 1: Summary of the Prediction Performance of MacroSimGNN and Graph Kernel on Testing-1 Dataset and Testing-2 Dataset.

Method	MacroSimGNN		Graph Kernel	
Testing Dataset	Testing-1	Testing-2	Testing-1	Testing-2
$\boldsymbol{R_s^2}$	$\mathbf{0.913 \pm 0.007}$	0.888 ± 0.007	-0.830	-0.801
$\boldsymbol{MAE_s}$	$\mathbf{0.0233 \pm 0.0008}$	0.0270 ± 0.0008	0.1400	0.1384
$\boldsymbol{MSE_s}$	$\mathbf{0.0015 \pm 0.0001}$	0.0020 ± 0.0001	0.0325	0.0321
$\boldsymbol{R_{NGED}^2}$	$\mathbf{0.953 \pm 0.003}$	0.935 ± 0.004	-0.931	-0.912
$\boldsymbol{MAE_{NGED}}$	$\mathbf{0.070 \pm 0.002}$	0.082 ± 0.003	0.518	0.516
$\boldsymbol{MSE_{NGED}}$	$\mathbf{0.0107 \pm 0.0008}$	0.0148 ± 0.0009	0.437	0.435
$\boldsymbol{R_{GED}^2}$	$\mathbf{0.968 \pm 0.002}$	0.958 ± 0.002	-0.131	-0.102
$\boldsymbol{MAE_{GED}}$	$\mathbf{0.41 \pm 0.01}$	0.47 ± 0.01	2.85	2.84
$\boldsymbol{MSE_{GED}}$	$\mathbf{0.37 \pm 0.03}$	0.52 ± 0.03	13.38	13.45

Impact of the Training Dataset Size

The impact of the Training dataset 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 dataset. 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 datasets, Testing-1 and Testing-2, are used for the evaluation process. For both the Testing-1 dataset and Testing-2 dataset, as the Training dataset size increases, the model's performance improves, as evidenced by increases in R^2 (Figure 7) and decreases in MAE (Figure 7) and MSE (Figure S2 in the Supporting Information) for GED, NGED, and s. Furthermore, the model's predictive performance stabilized when the training dataset size reached approximately 10^4. Beyond this point, increases in dataset size yielded diminishing returns in performance improvement but increased the cost of memory capacity and computational time. This trend was consistent across both testing datasets and all evaluation metrics.

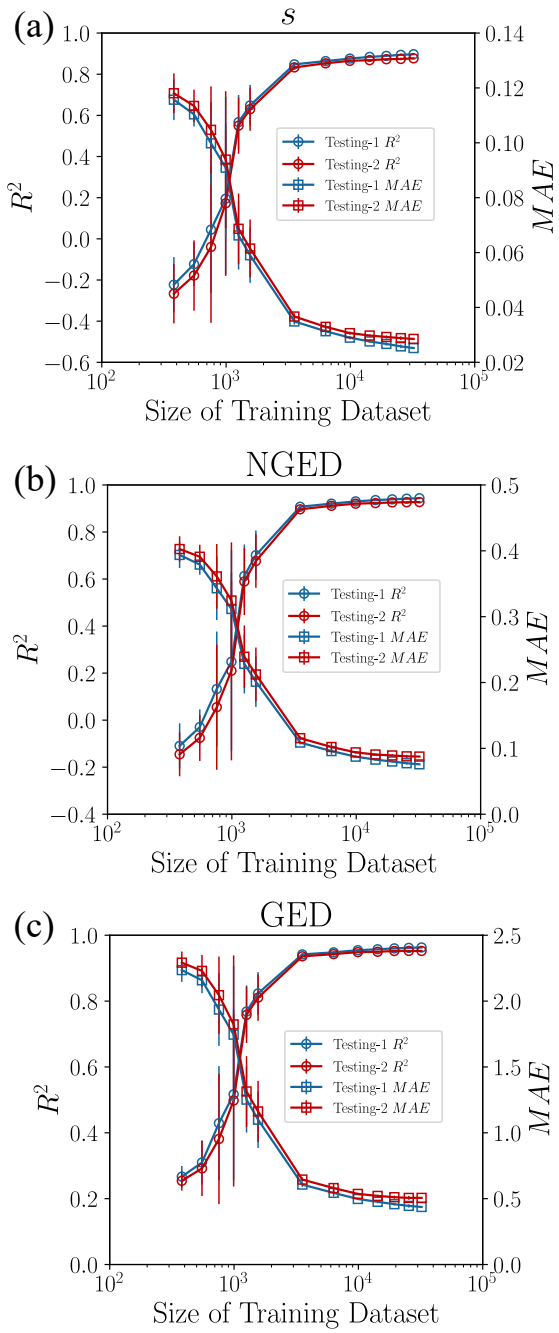


Figure 7: Impact of the training dataset size on the performance of MacroSimGNN predictions. (a) The left y-axis shows R^2 of s predictions on Testing-1 dataset (blue circles) and Testing-2 dataset (red circles); the right y-axis shows MAE of s predictions on Testing-1 dataset (blue squares) and Testing-2 dataset (red squares). (b) R^2 and MAE of NGED predictions. (c) R^2 and MAE of GED predictions. The error bar represents the standard deviation of the five randomly sampled subsets at each size. For both Testing-1 dataset and Testing-2, as the Training dataset size increases, the model's performance improves, as evidenced by increases in R^2 and decreases in MAE for s, NGED and GED. Furthermore, when the training dataset size reaches approximately 10^4,

increases in dataset size yielded diminishing returns in performance improvement of MacroSimGNN.

Computational Efficiency

As illustrated in Table 2, the computing speed for graph similarity calculation 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 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 2: Computing Efficiency for A* algorithm (Exact GED), MacroSimGNN and 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 dataset as an example. The glycan immunogenicity dataset comprises 470 non-immunogenic and 549 immunogenic glycans. MacroSimGNN is employed to obtain landmark distance embeddings[65, 66] for all glycans in this immunogenicity dataset. The indices of glycans have been reordered for intuitive visualization: indices 0-469 are non-immunogenic, and indices 470-1018 are immunogenic, as displayed in the pairwise dissimilarity ($d = 1 - s$) matrix of size 1019×1019 (Figure 8a). Noticeable differences exist between the non-immunogenic and immunogenic regions.

Each column of the dissimilarity matrix is a landmark distance embedding, which is a 1019-dimension 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 8b, showing that non-immunogenic and immunogenic glycans generally occupy distinct locations in the PCA space. Additionally, for supervised learning, the whole glycan immunogenicity dataset is divided into training and testing dataset with the ratio 4:1 (Specifically, 815 data for training the model and 204 data for the hold-out test dataset). Gaussian Process Classification using landmark distance embedding as inputs, predicts immunogenicity with 96% accuracy on the held-out test dataset, as shown in Figure 8c. The results of using NGED and GED for landmark embedding are illustrated in Figures 8d-f and Figures 8g-i, 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 94% accuracy on the held-out test dataset. 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.

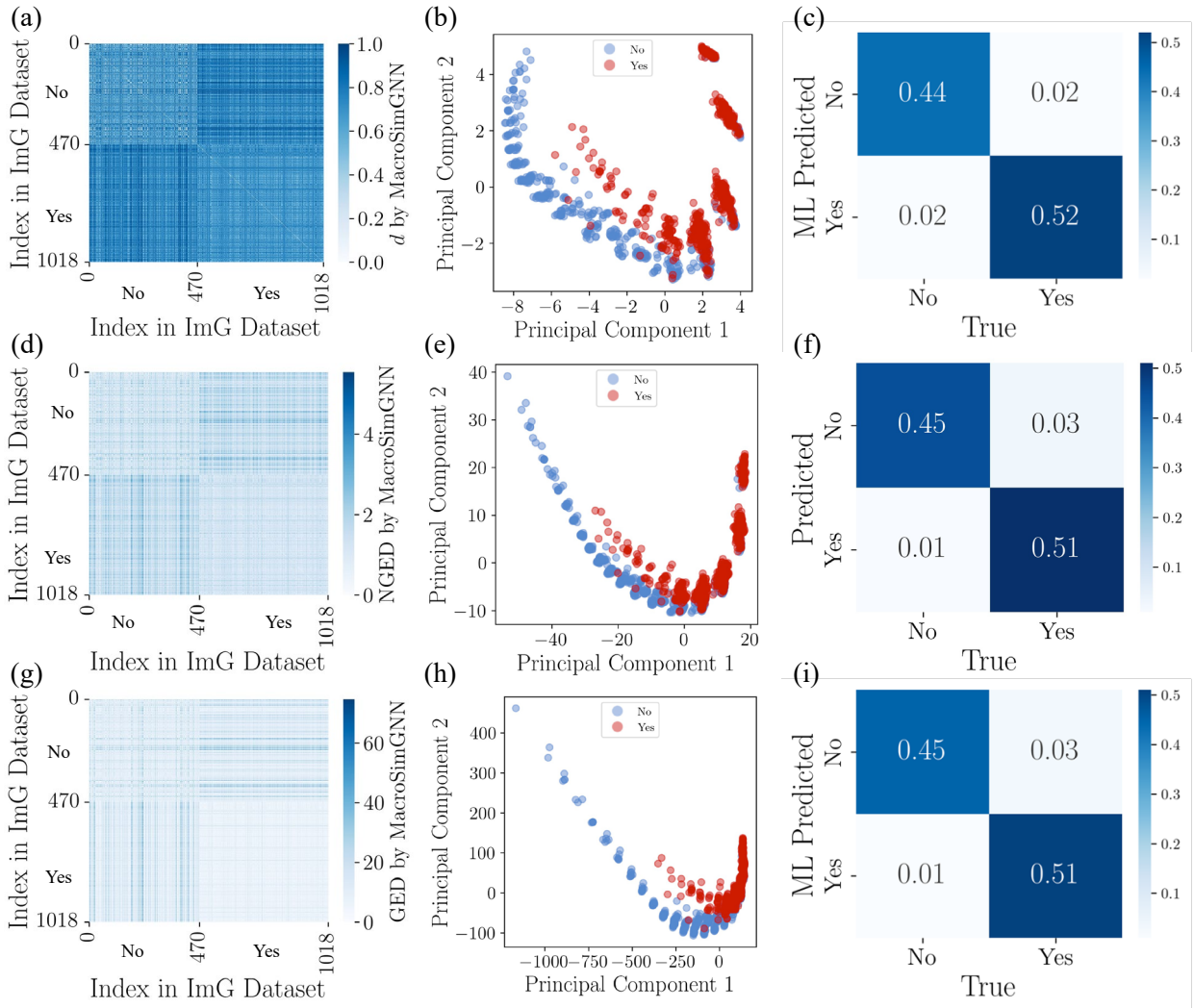


Figure 8: MacroSimGNN is applied to develop a landmark distance embedding for both unsupervised and supervised learning tasks for glycan immunogenicity (ImG) dataset. (a) Pairwise dissimilarity ($d = 1 - s$) matrix where indices 0-469 are non-immunogenic and 470-1018 are immunogenic. The index of glycans has been reordered for intuitive visualization. Noticeable differences exist between the non-immunogenic and immunogenic regions. (b) Dimension reduction results from PCA show that non-immunogenic (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% accuracy on the hold-out test dataset. (d) (e) (f) are the results of using NGED as landmark distance embedding, and (g) (h) (i) are the results of GED as landmark distance embedding. Respectively, they are similar to the results of using dissimilarity.

Conclusion

This study introduces MacroSimGNN, a graph neural network model designed to accelerate pairwise graph similarity calculations 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. 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.

Code Availability

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.

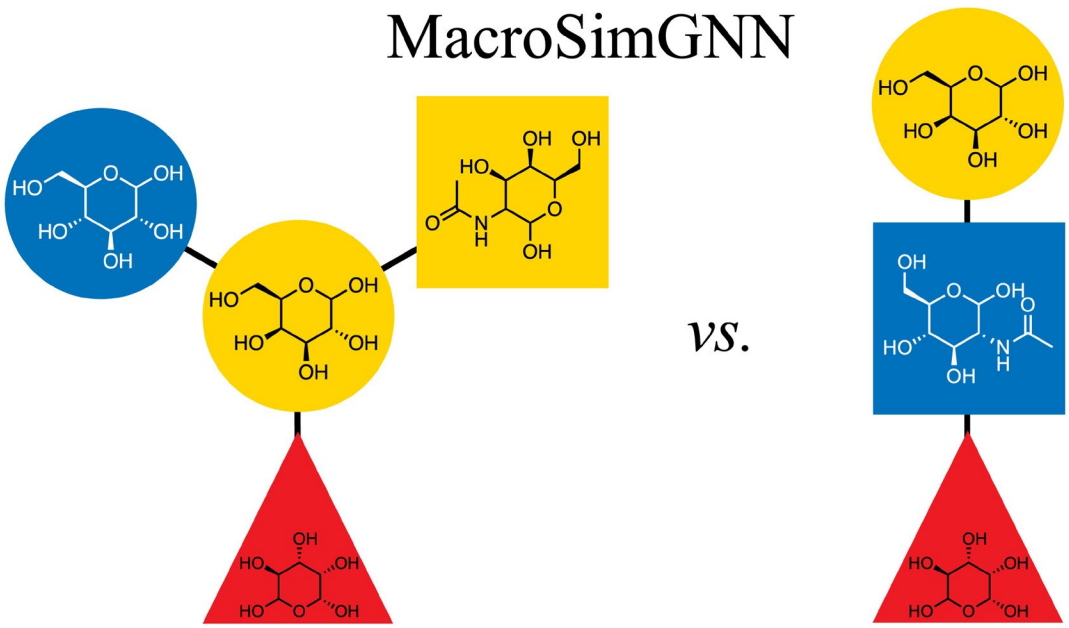
Acknowledgment

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.

Notes

Certain equipment, instruments, software, or materials, commercial or non-commercial, 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.

TOC Graph



References

(1) Rowan, S. J. 100th Anniversary of Macromolecular Science Viewpoints. *ACS Macro Letters* **2021**, *10* (4), 466-468. DOI: 10.1021/acsmacrolett.1c00175.
(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 Letters* **2020**, *9* (9), 1292-1302. DOI: 10.1021/acsmacrolett.0c00488.
(3) Mohapatra, S.; An, J.; Gómez-Bombarelli, R. *Graph attribution methods applied to understanding immunogenicity in glycans*.
(4) Varki, A. Biological roles of glycans. *Glycobiology* **2016**, *27* (1), 3-49. DOI: 10.1093/glycob/cww086 (acccessed 6/28/2024).
(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**. DOI: 10.1038/s41586-023-05993-x (acccessed 2023-05-01T16:11:15).
(6) Jones, S.; Thornton, J. M. Principles of protein-protein interactions. *Proceedings of the National Academy of Sciences* **1996**, *93* (1), 13-20. DOI: doi:10.1073/pnas.93.1.13.
(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 Research* **2021**, *49* (D1), D404-D411. DOI: 10.1093/nar/gkaa1021.
(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. *Science China Chemistry* **2021**, *64* (2), 171-203. DOI: 10.1007/s11426-020-9864-7.
(9) Opalinska, J. B.; Gewirtz, A. M. Nucleic-acid therapeutics: basic principles and recent applications. *Nature Reviews Drug Discovery* **2002**, *1* (7), 503-514. DOI: 10.1038/nrd837.
(10) Eschenmoser, A. Chemical Etiology of Nucleic Acid Structure. *Science* **1999**, *284* (5423), 2118-2124. DOI: doi:10.1126/science.284.5423.2118.
(11) Provin, A. P.; Regina de Aguiar Dutra, A.; Machado, M. M.; Vieira Cubas, A. L. New materials for clothing: Rethinking possibilities through a sustainability approach - A review. *Journal of Cleaner Production* **2021**, *282*, 124444-124444. DOI: 10.1016/j.jclepro.2020.124444.
(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. *Journal of Polymer Science Part B-Polymer Physics* **2010**, *48* (15), 1685-1718. DOI: 10.1002/polb.22037.
(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. *Chemical Reviews* **2020**, *120* (15), 7642-7707. DOI: 10.1021/acs.chemrev.0c00345.
(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. DOI: 10.1021/ma1010099.
(15) Yadav, R.; Tirumali, M.; Wang, X.; Naebe, M.; Kandasubramanian, B. Polymer composite for antistatic application in aerospace. *Defence Technology* **2020**, *16* (1), 107-118. DOI: 10.1016/j.dt.2019.04.008.
(16) Pendhari, S. S.; Kant, T.; Desai, Y. M. Application of polymer composites in civil construction: A general review. *Composite Structures* **2008**, *84* (2), 114-124. DOI: https://doi.org/10.1016/j.compstruct.2007.06.007.
(17) Stenzel, M. H. Glycopolymers for Drug Delivery: Opportunities and Challenges. *Macromolecules* **2022**, *55* (12), 4867-4890. DOI: 10.1021/acs.macromol.2c00557.
(18) Le, T.; Epa, V. C.; Burden, F. R.; Winkler, D. A. Quantitative Structure-Property Relationship Modeling of Diverse Materials Properties. *Chemical Reviews* **2012**, *112* (5), 2889-2919. DOI: 10.1021/cr200066h.
(19) Ma, R. M.; Liu, Z. Y.; Zhang, Q. W.; Liu, Z. Y.; Luo, T. F. Evaluating Polymer Representations via Quantifying Structure-Property Relationships. *Journal of Chemical Information and Modeling* **2019**, *59* (7), 3110-3119. DOI: 10.1021/acs.jcim.9b00358.
(20) Otsuka, S.; Kuwajima, I.; Hosoya, J.; Xu, Y.; Yamazaki, M. PoLyInfo: Polymer Database for Polymeric Materials Design. In *2011 International Conference on Emerging Intelligent Data and Web Technologies*, 2011/9//, 2011; IEEE: pp 22-29. DOI: 10.1109/EIDWT.2011.13.
(21) Ma, R. M.; Luo, T. F. PI1M: A Benchmark Database for Polymer Informatics. *Journal of Chemical Information and Modeling* **2020**, *60* (10), 4684-4690. DOI: 10.1021/acs.jcim.0c00726.
(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. *Journal of Applied Physics* **2020**, *128* (17). DOI: 10.1063/5.0023759.
(23) Kim, S.; Schroeder, C. M.; Jackson, N. E. Open Macromolecular Genome: Generative Design of Synthetically Accessible Polymers. *Acs Polymers Au*. DOI: 10.1021/acspolymersau.3c00003.
(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 9/21/2023).
(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 Materials* **2016**, *4* (5). DOI: 10.1063/1.4943679.
(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 Letters* **2020**, *9* (8), 1086-1094. DOI: 10.1021/acsmacrolett.0c00264.
(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. *Chemistry of Materials* **2021**, *33* (17), 7008-7016. DOI: 10.1021/acs.chemmater.1c02061.
(28) Kuenneth, C.; Ramprasad, R. polyBERT: a chemical language model to enable fully machine-driven ultrafast polymer informatics. *Nature Communications* **2023**, *14* (1), 4099. DOI: 10.1038/s41467-023-39868-6.
(29) Webb, M. A.; Jackson, N. E.; Gil, P. S.; de Pablo, J. J. Targeted sequence design within the coarse-grained polymer genome. *Science Advances* **2020**, *6* (43). DOI: 10.1126/sciadv.abc6216.
(30) Patel, R. A.; Borca, C. H.; Webb, M. A. Featurization strategies for polymer sequence or composition design by machine learning. *Molecular Systems Design & Engineering* **2022**, *7* (6), 661-676. DOI: 10.1039/D1ME00160D.
(31) Zhang, Y.; Xu, X. J. Machine learning glass transition temperature of polymers. *Heliyon* **2020**, *6* (10), 7. DOI: 10.1016/j.heliyon.2020.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 Applied Polymer Materials* **2020**, *2* (8), 3107-3113. DOI: 10.1021/acsapm.0c00273.
(33) Tao, L.; Varshney, V.; Li, Y. Benchmarking Machine Learning Models for Polymer Informatics: An Example of Glass Transition Temperature. *Journal of Chemical Information and Modeling* **2021**, *61* (11), 5395-5413. DOI: 10.1021/acs.jcim.1c01031.
(34) Tao, L.; Chen, G.; Li, Y. Machine learning discovery of high-temperature polymers. *Patterns* **2021**, *2* (4), 15. DOI: 10.1016/j.patter.2021.100225.
(35) Chen, L.; Pilania, G.; Batra, R.; Huan, T. D.; Kim, C.; Kuenneth, C.; Ramprasad, R. Polymer informatics: Current status and critical next steps. *Materials Science and Engineering: R: Reports* **2021**, *144*, 100595-100595. DOI: 10.1016/j.mser.2020.100595.
(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. DOI: 10.1039/D1SM01012C.
(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 Applied Materials & Interfaces* **2022**, *14* (32), 37161-37169. DOI: 10.1021/acsami.2c08891.

(38) Gormley, A. J.; Webb, M. A. Machine learning in combinatorial polymer chemistry. *Nature Reviews Materials* **2021**, *6* (8), 642-644. DOI: 10.1038/s41578-021-00282-3.
(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. *Advanced Materials* **2022**, *34* (30), 2201809. DOI: 10.1002/adma.202201809.
(40) Patra, T. K. Data-Driven Methods for Accelerating Polymer Design. *Acs Polymers Au* **2022**, *2* (1), 8-26. DOI: 10.1021/acspolymersau.1c00035.
(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 Letters* **2021**, *10* (11), 1339-1345. DOI: 10.1021/acsmacrolett.1c00521.
(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. DOI: 10.1021/acs.macromol.2c02165.
(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 Central Science* **2022**, *8* (7), 996-1007. DOI: 10.1021/acscentsci.2c00382.
(44) Yang, J.; Tao, L.; He, J.; McCutcheon, J. R.; Li, Y. Machine learning enables interpretable discovery of innovative polymers for gas separation membranes. *Science Advances* **2022**, *8* (29), 9545-9545. DOI: 10.1126/SCIADV.ABN9545.
(45) Smith, T. F.; Waterman, M. S. Identification of common molecular subsequences. *Journal of Molecular Biology* **1981**, *147* (1), 195-197. DOI: 10.1016/0022-2836(81)90087-5.
(46) Eddy, S. R. Where did the BLOSUM62 alignment score matrix come from? *Nature Biotechnology* **2004**, *22* (8), 1035-1036. DOI: 10.1038/nbt0804-1035.
(47) Mohapatra, S.; An, J.; Gomez-Bombarelli, R. Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning. *Machine Learning-Science and Technology* **2022**, *3* (1), 11. DOI: 10.1088/2632-2153/ac545e.
(48) Wu, W.; Wang, W.; Li, J. Star polymers: Advances in biomedical applications. *Progress in Polymer Science* **2015**, *46*, 55-85. DOI: 10.1016/j.progpolymsci.2015.02.002.
(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. DOI: 10.1039/C4PY00484A.
(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. *Chemical Reviews* **2021**, *121* (8), 5042-5092. DOI: 10.1021/acs.chemrev.0c01304.
(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). DOI: 10.1093/bioinformatics/btad298 (acccessed 7/4/2024).
(52) Wang, Y.; Kalscheur, J.; Ebikade, E.; Li, Q.; Vlachos, D. G. LigninGraphs: lignin structure determination with multiscale graph modeling. *Journal of Cheminformatics* **2022**, *14* (1), 43. DOI: 10.1186/s13321-022-00627-2.
(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**. DOI: 10.1021/acs.macromol.3c00994.
(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 Polymers Au* **2024**. DOI: 10.1021/acspolymersau.3c00029.
(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. DOI: 10.1021/acs.macromol.3c00761.
(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 Transactions on Neural Networks and Learning Systems* **2021**, *32* (1), 4-24, Article. DOI: 10.1109/tnnls.2020.2978386.
(57) Siglidis, G.; Nikolentzos, G.; Limnios, S.; Giatsidis, C.; Skianis, K.; Vazirgiannis, M. GraKeL: A Graph Kernel Library in Python. *Journal of Machine Learning Research* **2020**, *21* (54), 1-5.
(58) Neumann, M.; Garnett, R.; Bauckhage, C.; Kersting, K. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning* **2016**, *102* (2), 209-245. DOI: 10.1007/s10994-015-5517-9.
(59) Kriege, N. M.; Johansson, F. D.; Morris, C. A survey on graph kernels. *Applied Network Science* **2020**, *5* (1), 6. DOI: 10.1007/s41109-019-0195-3.
(60) Sanfeliu, A.; Fu, K.-S. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics* **1983**, *SMC-13* (3), 353-362. DOI: 10.1109/TSMC.1983.6313167.
(61) Blumenthal, D. B.; Gamper, J. On the exact computation of the graph edit distance. *Pattern Recognition Letters* **2020**, *134*, 46-57. DOI: 10.1016/j.patrec.2018.05.002.
(62) Bai, Y. S.; Ding, H.; Bian, S.; Chen, T.; Sun, Y. Z.; Wang, W.; Acm. SimGNN: A Neural Network Approach to Fast Graph Similarity Computation. In *12th ACM International Conference on Web Search and Data Mining (WSDM)*, Melbourne, AUSTRALIA, Feb 11-15, 2019; Assoc Computing Machinery: NEW YORK, 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 Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45* (6), 6984-7000. DOI: 10.1109/TPAMI.2020.3005590.
(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 Transactions on Pattern Analysis and Machine Intelligence* **2022**, *44* (9), 5261-5279. DOI: 10.1109/TPAMI.2021.3078053.
(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. *Journal of Enzyme Inhibition and Medicinal Chemistry* **2023**, *38* (1), 24-35. DOI: 10.1080/14756366.2022.2132486.
(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. *Journal of Chemical Information and Modeling* **2022**, *62* (15), 3551-3564. DOI: 10.1021/acs.jcim.2c00229.
(67) Socher, R.; Chen, D.; Manning, C. D.; Ng, A. Y. Reasoning with neural tensor networks for knowledge base completion. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada; 2013.

(68) Greenacre, M.; Groenen, P. J. F.; Hastie, T.; D'Enza, A. I.; Markos, A.; Tuzhilina, E. Principal component analysis. *Nature Reviews Methods Primers* **2022**, *2* (1), 1-21, ReviewPaper. DOI: doi:10.1038/s43586-022-00184-w.
(69) 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. *Journal of Chemical Education* **2020**, *97* (6), 1666-1676. DOI: 10.1021/acs.jchemed.9b00924.
(70) Héberger, K.; Milczewska, K.; Voelkel, A. Principal component analysis of polymer–solvent and filler–solvent interactions by inverse gas chromatography. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **2005**, *260* (1), 29-37. DOI: 10.1016/j.colsurfa.2005.02.029.
(71) Banerjee, A.; Hsu, H.-P.; Kremer, K.; Kukharenko, O. Data-Driven Identification and Analysis of the Glass Transition in Polymer Melts. *ACS Macro Letters* **2023**, *12* (6), 679-684. DOI: 10.1021/acsmacrolett.2c00749.
(72) 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. *Journal of Machine Learning Research 12*, 2825-2830.
(73) Deringer, V. L.; Bartók, A. P.; Bernstein, N.; Wilkins, D. M.; Ceriotti, M.; Csányi, G. Gaussian Process Regression for Materials and Molecules. *Chemical Reviews* **2021**, *121* (16), 10073-10141. DOI: 10.1021/acs.chemrev.1c00022.
(74) 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.
(75) Chen, Z.; Li, D.; Liu, J.; Gao, K. Application of Gaussian processes and transfer learning to prediction and analysis of polymer properties. *Computational Materials Science* **2023**, *216*, 111859. DOI: 10.1016/j.commatsci.2022.111859.
(76) Obrezanova, O.; Segall, M. D. Gaussian Processes for Classification: QSAR Modeling of ADMET and Target Activity. *Journal of Chemical Information and Modeling* **2010**, *50* (6), 1053-1061. DOI: 10.1021/ci900406x.
(77) Bojar, D.; Powers, R. K.; Camacho, D. M.; Collins, J. J. Deep-Learning Resources for Studying Glycan-Mediated Host-Microbe Interactions. *Cell Host and Microbe* **2021**, *29* (1), 132-144.e133. DOI: 10.1016/j.chom.2020.10.004.
(78) Abu-Aisheh, Z.; Raveaux, R.; Ramel, J. Y.; Martineau, P. An exact graph edit distance algorithm for solving pattern recognition problems. *ICPRAM 2015 - 4th International Conference on Pattern Recognition Applications and Methods, Proceedings* **2015**, *1*, 271-278. DOI: 10.5220/0005209202710278.
(79) *NetworkX*. 2024. https://networkx.org/ (accessed 10/12/2023).
(80) Bajusz, D.; Rácz, A.; Héberger, K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics* **2015**, *7* (1), 20-20. DOI: 10.1186/s13321-015-0069-3.