BioData Mining

METHODOLOGY Open Access

MultiChem: predicting chemical properties using multi-view graph attention network

Heesang Moon[1] and Mina Rho[1,2,3*]

*Correspondence:
minarho@hanyang.ac.kr

[1] Department of Computer Science, Hanyang University, Seoul, Republic of Korea
[2] Department of Artificial Intelligence, Seoul, Republic of Korea
[3] Department of Biomedical Informatics, Hanyang University, Seoul, Republic of Korea

Abstract

Background: Understanding the molecular properties of chemical compounds is essential for identifying potential candidates or ensuring safety in drug discovery. However, exploring the vast chemical space is time-consuming and costly, necessitating the development of time-efficient and cost-effective computational methods. Recent advances in deep learning approaches have offered deeper insights into molecular structures. Leveraging this progress, we developed a novel multi-view learning model.

Results: We introduce a graph-integrated model that captures both local and global structural features of chemical compounds. In our model, graph attention layers are employed to effectively capture essential local structures by jointly considering atom and bond features, while multi-head attention layers extract important global features. We evaluated our model on nine MoleculeNet datasets, encompassing both classification and regression tasks, and compared its performance with state-of-the-art methods. Our model achieved an average area under the receiver operating characteristic (AUROC) of 0.822 and a root mean squared error (RMSE) of 1.133, representing a 3% improvement in AUROC and a 7% improvement in RMSE over state-of-the-art models in extensive seed testing.

Conclusion: MultiChem highlights the importance of integrating both local and global structural information in predicting molecular properties, while also assessing the stability of the models across multiple datasets using various random seed values.

Implementation: The codes are available at https://github.com/DMnBI/MultiChem.

Introduction

Accurate characterization of the molecular properties of chemical compounds is crucial for identifying lead candidates and evaluating cellular and tissue-level interactions in drug development. With over 100 million molecules available, experimentally screening desirable compounds within this vast chemical space is challenging, time-consuming, and costly. In this context, in silico evaluation provides a more efficient alternative, contingent on the development of accurate predictive models.

In recent years, machine learning methods have become widely adopted in drug discovery and molecular property prediction [43, 66]. These methods, grounded in

quantitative structure–activity relationship (QSAR) principles, predict molecular activity based on structural similarities by carefully selecting relevant variables and inference functions [21]. Traditional machine learning models, such as support vector machine [7] and random forests [2], have demonstrated strong performance in estimating relationships between molecular structures and their properties.

In QSAR modeling, molecular fingerprints have become a standard for representing molecular structures [43, 66]. These fingerprints capture features such as electronic, geometric, and steric properties. Commonly used fingerprints, like PubChem fingerprint [30], MACCS Key [8], and extended-connectivity fingerprints (ECFP) [49], can be generated efficiently using tools like PaDel [65], RDKit [32], and CDK [53]. However, these approaches depend on predefined substructures, which may overlook other relevant molecular features [26], highlighting the need for more comprehensive molecular representations.

Deep learning has recently gained prominence across various fields due to its ability to automatically extract features without manual intervention [13]. In drug discovery, graph-based and sequence-based neural networks have shown particular promise by capturing a broader range of substructures than traditional fingerprints [27, 28, 37, 40, 44, 45, 56, 59, 67, 70]. Graph-based models are adept at distinguishing locally positioned substructures, while sequence-based models capture global patterns from distant substructures.

Several graph neural networks (GNNs) have been introduced for molecular property prediction, including graph convolutional network (GCN) [31], message-passing neural network (MPNN) [19], and graph attention network (GAT) [3, 57]. These models focus on individual nodes within molecular graphs to accurately capture local structural features. Studies have shown that GCNs and MPNNs perform comparably to traditional machine learning approaches [6, 9, 19, 29, 31, 39, 51, 55]. GATs and their variants, incorporating attention mechanisms, have shown better performance by assigning varying influence probabilities within the graph structure [34, 62, 68].

Building on the success of atom-centered GNNs, several variant models have been developed to enhance prediction accuracy [15, 36, 38, 54, 64, 69]. Among these, the directional message-passing neural network (DMPNN) [63] is notable for its focus on bond information, treating atoms and bonds are equally important to better differentiate non-isomorphic structures. Unlike atom-centered GNNs, the bond-centered DMPNN utilizes a line graph where the atoms and bonds are assigned to edges and nodes, respectively [18, 60, 63]. Inspired by the success of both atom- and bond-centered GNNs, several studies have attempted to combine atom and bond representations in a single framework [5, 15, 40, 52, 59]. In these studies, either a bond-centered GNN was added to an atom-centered GNN, or both GNNs were used together to exchange information. However, these approaches perform simple summation or averaging to represent atoms and bonds rather than employing more sophisticated attention, potentially limiting their effectiveness.

Sequence-based models have also been employed in molecular property prediction [20, 25, 50, 58]. These models primarily use SMILES sequences to represent molecular structures, enabling them to extract global structural features from entire sequences. Notably, BERT has been adapted for molecular property prediction using fully connected graph

approaches [58]. Similar to GNNs, sequence-based models often outperform traditional methods.

Data scarcity is a common challenge in molecular property prediction, leading to adoption of multi-task learning and transfer learning to address this issue. Multi-task learning improves performance by sharing information across related tasks [43, 61]. Transfer learning, widely used in both graph- and sequence-based models, enables the leveraging of knowledge from larger datasets to enhance predictive accuracy on smaller datasets [16, 35, 50, 58].

In this work, we developed a model that generates a comprehensive molecular representation by simultaneously capturing local and global structural information. Our approach incorporates two graph encoders to effectively capture local information from atoms and bonds. These local features are jointly aggregated using a graph attention mechanism, which represents a key contribution of our study. Unlike traditional aggregation methods [34, 57, 62], our mechanism more effectively captures structural information from atoms and bonds, making it particularly well-suited for molecular property prediction. To complement the local features, we applied a multi-head attention mechanism to capture global relationships between distant substructures across diverse molecular topologies. By directly applying multi-head attention to the local features from our graph encoders, our model emphasizes both substructural relationships while simultaneously extracting global relationships. This multi-view approach integrates local and global information, enabling a more refined representation of molecular structures. When we assessed the impact of individual components within our model, we observed AUROC improvements ranging from 1.0% to 5.1% when comparing the model to the other models lacking specific components. In addition, the final model with ensemble method showed a 2.3% increase in AUROC score.

In our experiments, several models showed substantial performance differences between the different datasets. These inconsistencies may be due to structural differences in the scaffold distributions of the datasets, suggesting that each model focused on different molecular regions during training. To address the challenges of data scarcity and imbalance, we also incorporated an ensemble method, which has also shown promising results in the biological fields [12, 14]. Ensemble methods reduce variance and bias, mitigating risks of overfitting. In particular, bagging in our method promotes model diversity by training on data subsets with varied distributions, improving generalization and stability by reducing variance across individual models.

We evaluated our method on nine benchmark datasets from the MoleculeNet, including both classification and regression tasks. Compared to state-of-the-art methods [16, 34, 35, 40, 50] on golden standard datasets, our model achieved better or comparable performances, with an average area under the receiver operating characteristic (AUROC) of 0.822 and a root mean squared error (RMSE) of 1.133, reflecting improvements of 3% in AUROC and 7% in RMSE over current leading methods.

Materials and methods

Datasets

We used classification and regression datasets from MoleculeNet [61] using the DeepChem [47] library. The classification datasets include six tasks of BBBP [42], Tox21 [23],

ToxCast [48], SIDER [47], ClinTox [61], and BACE. The regression datasets include three tasks of ESOL, FreeSolv, and Lipophilicity regarding physical chemistry.

The BBBP dataset provides information on molecular penetration of the blood–brain barrier [42]. Tox21 includes qualitative screening data for nuclear receptor (NR) and stress response (SR) assays [23]. ToxCast comprises extensive in vitro high-throughput screening data [48]. SIDER contains adverse drug reaction data classified by MedDRA [47], and ClinTox includes FDA-approved drugs alongside those failed due to toxicity issues [61]. The BACE dataset provides biophysical properties related to binding for inhibitors of human beta-secretase 1 (BACE-1). The ESOL dataset contains 1,128 compounds with water solubility. FreeSolv provides compounds with experimental hydration-free energy in water. The Lipophilicity dataset contains experimental octanol/water distribution coefficient values. In classification tasks, each task classifies compounds as active, inactive, and inconclusive. For this study, we focus exclusively on the active and inactive outcomes to maintain accuracy and reliability. Detailed configurations of each dataset are provided in Table 1.

Feature initialization

Our model utilizes two graph forms: atom graph and bond graph (Fig. 1A), designed for the atom-centered and bond-centered sub-models, respectively. The atom graph represents atoms as nodes and bonds as edges, with the adjacency matrix indicating the bond presence. The bond graph assigns bonds as nodes and atoms as edges, with the adjacency matrix indicating direct connections between two bonds via an atom. This dual-input approach ensures our model equally considers both atoms and bonds.

To initialize features, we represented the molecule with three matrices (node vectors, edge vectors, and adjacency matrix), a common method in deep learning for drug discovery [63]. Node vectors are based on atomic properties such as atom type, the number of bonds, and mass. Edge vectors include bond properties like bond type, ring presence, and conjugation. Properties are transformed into feature vectors using one-hot encoding or scaling, resulting in a node vector of length 127 and an edge vector of length 12. For example, atom type and mass are converted to a vector of length 100 by one-hot encoding and a vector of length one by scaling, respectively. Bond type is transformed into a vector of length four by one-hot encoding. Feature configurations are detailed in Table 2, with properties extracted from the SMILES string using RDKit [32].

Table 1 Details of the datasets

Dataset	#Tasks	#Compounds	#Actives	#Inactives	#Atoms	#Bonds
BBBP	1	2039	1560	479	49,068	52,921
Tox21	12	7831	5862	72,084	145,459	151,095
ToxCast	617	8576	126,651	1,407,009	161,088	165,178
SIDER	27	1427	21,868	16,661	48,006	50,456
ClinTox	2	1478	1496	1460	38,661	41,209
BACE	1	1513	691	822	51,577	55,768
FreeSolv	1	642	-	-	5600	5385
ESOL	1	1128	-	-	14,991	15,428
Lipo	1	4200	-	-	113,568	123,899

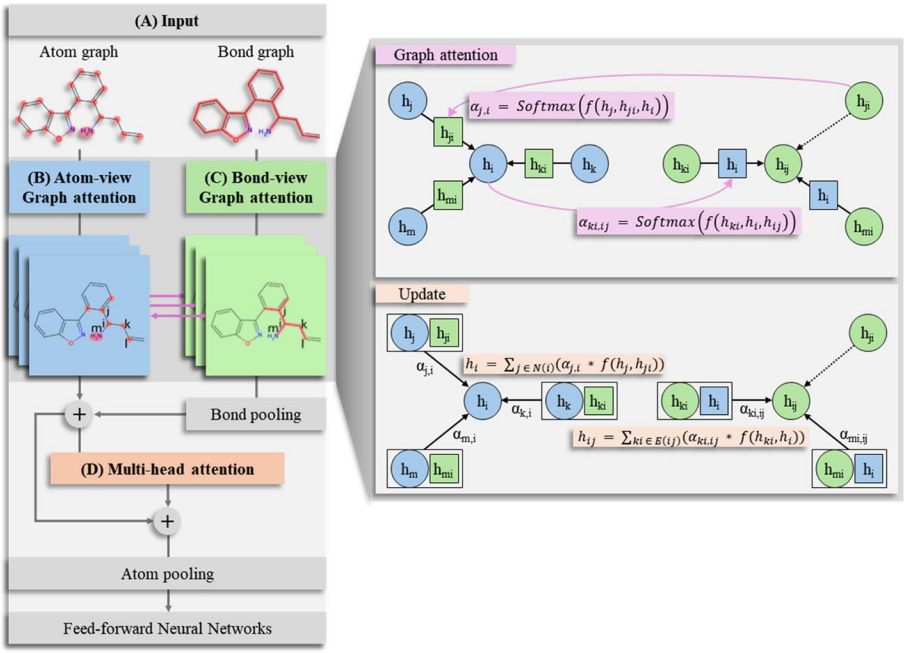

Fig. 1 MultiChem: a practical tool for molecular property prediction. **A** Inputs. We employed two molecular graphs for our model. One is the Atom graph, which assigns atoms and bonds as nodes and edges. The other is the Bond graph, which allocates bonds and atoms as nodes and edges. **B** GNN for atom. We modified the graph attention network to incorporate the bond information from the other GNN and used it for the Atom graph. **C** GNN for bond. Similarly, we adapted the graph attention network to reflect the atom information from the GNN for the atom and exploited it to the Bond graph. **D** Attention. We adopted a self-attention mechanism between the nodes in the molecule to capture the long-distance features

Table 2 Feature initialization

	Feature	Value	Encoding
Atom	Atom type	0 to 100	one-hot encoding
	The number of Bonds	0, 1, 2, 3, 4, 5	one-hot / real number
	Formal Charge	−2, −1, 0, 1, 2	one-hot / real number
	Chirality	unspecified, tetrahedral_cw, tetrahedral_ccw, other	one-hot encoding
	The number of Hs	0, 1, 2, 3, 4	one-hot / real number
	Hybridization	sp, sp2, sp3, sp3d, sp3d2	one-hot encoding
	Aromaticity	0, 1	one-hot encoding
	Atomic mass	mass	real number
Bond	Bond type	single, double, triple, aromatic	one-hot encoding
	Conjugated	0, 1	one-hot encoding
	In ring	0, 1	one-hot encoding
	Stereo	one, any, e, z, cis, trans	one-hot encoding

Model construction

In the MultiChem model, we incorporate three key components: a graph attention mechanism to integrate atom and bond information, a multi-head attention mechanism to capture both local and global features, and an ensemble method to enhance

model stability. Since molecules are composed of atoms connected by covalent bonds, accurately predicting molecular properties requires models that account for both atomic and bonding information. Recent studies [5, 15, 40, 52, 59] have shown that models integrating both atomic and bond information provide more accurate predictions and richer representations compared to those focusing primarily on atoms and treating bonds merely as connections.

In our approach, we incorporate distinct graphs for atoms and bonds and iteratively updating both through a process of mutual reinforcement, where nodes are updated based on edge information and edges are updated based on node information. Further, we advance the integration of atom and bond information by employing a graph attention mechanism, which is known for its flexibility in learning structural information by assigning different weights to neighboring nodes, thus identifying crucial substructures. Our method uses two graph attention encoders—one for atoms and one for bonds–that are interconnected by sharing certain hidden states of atoms and bonds, updated using attention weights derived from neighboring atom and bond hidden states.

After extracting local features with the graph attention encoders, we apply a multi-head attention mechanism to the combined features, aiming to improve predictive power. Multi-head attention is recognized for its reliability and stability compared to single attention mechanisms. While GNNs are typically limited to a predefined number of hops, multi-head attention allows for capturing useful information between distant nodes. By applying multi-head attention to the outputs from our local graph-based module, the model can extract global features from distant parts of the graph, complementing the local features provided by the graph encoders. Further, to integrate local and global features effectively, we used a residual connection between the local and global modules, which helps prevent overfitting in the multi-head attention layers and preserves earlier local information.

To further improve robustness and performance, we adopted bagging, an ensemble method, during model training and inference. Ensemble methods like bagging reduce variance and bias, mitigating risks of overfitting and underfitting. In particular, bagging in our method promotes model diversity by training on data subsets with varied distributions, improving generalization and stability by reducing variance across individual models. Bagging is well-suited for our method because we can generate diverse training data by employing *balanced scaffold splitting method*, as described in later section. For final inference in both classification and regression tasks, we used soft voting, which generally produces more reliable results than hard voting by averaging the predictions from multiple models.

This combined approach—graph attention for interactive atom-bond representation, multi-head attention for global feature extraction, and ensemble bagging for robustness–provides a comprehensive framework for accurate molecular property prediction.

Sub-model for the node

The node sub-model (Fig. 1B) is based on the graph attention network [2, 57], applying attention to connected nodes in the message-passing layers of the message-passing neural network. Message-passing involves exchanging information (i.e., message) between two connected nodes, and we incorporated the attention method into this process.

The initial hidden vector of the ith node as h_i^0, is derived from the initial node vector x_i through the function f_{init} (Eq. 1), which includes learning parameters, a bias, and the exponential linear unit (ELU) activation function.

The message vector $m_{j,i}^l$ in the message-passing layer is derived from the node vectors h_j^{l-1}, h_i^{l-1}, and the edge vector h_{ji}^{l-1} from the previous layer using the function $f_{message}$ (Eq. 2). The hidden vector h_{ji}^l is derived from the directional edge from the jth node to the ith node in the lth hidden layer in the edge sub-model, described in the next section. The attention coefficient $\alpha_{j,i}^l$ (Eq. 3) is calculated by applying the *softmax* function to the message vectors $m_{j,i}^l$, with j in the range of the neighbors of the ith node (N(i)). The node update function (Eq. 4) is formulated to obtain the node vector h_i^l, considering the edge vector h_{ji}^{l-1}, the attention score $\alpha_{j,i}^l$, and the node vector h_j^{l-1}. Finally, multi-head attention with k heads, a more stable method than single attention, is applied to the node update function (Eq. 5).

$$h_i^0 = f_{init}(x_i) \tag{1}$$

$$m_{j,i}^l = f_{message}\left(h_j^{l-1}, h_i^{l-1}, h_{ji}^{l-1}\right) \tag{2}$$

$$\alpha_{j,i}^l = Softmax_{j\in N(i)}\left(m_{j,i}^l\right) \tag{3}$$

$$h_i^l = \sum\nolimits_{j\in N(i)} \alpha_{j,i}^l * f_{update}(h_j^{l-1}, h_{ji}^{l-1}) \tag{4}$$

$$h_i^l = \frac{1}{K}\sum_{k=1}^{K}\sum\nolimits_{j\in N(i)} \alpha_{j,i}^{l,k} * f_{update}^k(h_j^{l-1,k}, h_{ji}^{l-1,k}) \tag{5}$$

Sub-model for the edge

The edge sub-model (Fig. 1C), also based on the graph attention network, was constructed similarly to the node sub-model. We applied the attention mechanism to the directional message-passing neural network [63] to incorporate hidden states and messages for the edges, creating a novel model. By focusing on bonds, our model captures features distinct from those obtained by the node model, making it valuable for bond-centric characteristics. This initial hidden vector of the bond between the ith and jth nodes, h_{ij}^0, is derived from the initial edge vector e_{ij}, node vector x_i, and node vector x_j through the function f_{init} (Eq. 6). The hidden vector of the ith node in the lth hidden layer, h_i^l, is inherited from the node sub-model.

With the initial states defined, we constructed the novel message function $f_{message}$. The message $m_{ki,ij}^l$ is calculated using the edge h_{ki}^{l-1}, the edge h_{ij}^{l-1}, and the node h_i^{l-1} (Eq. 7). Here, ki represents the edge from the kth node to the ith node, ij represents the edge from the ith node to the jth node, and l denotes the layer depth. These messages derive the attention score $\alpha_{ki,ij}^l$ through a *softmax* function (Eq. 8). This function operates among the edges

(E(*ij*)), directly connected to the edge *ij*. The aggregating function (Eq. 9) leverages attention scores, edges, and nodes. Finally, we introduced the multi-head attention with n heads to this sub-model (Eq. 10).

$$h_{ij}^{0} = f_{init}\left(x_i, x_j, e_{ij}\right) \tag{6}$$

$$m_{ki,ij}^{l} = f_{message}\left(h_{ki}^{l-1}, h_{ij}^{l-1}, h_{i}^{l-1}\right) \tag{7}$$

$$\alpha_{ki,ij}^{l} = Softmax_{ki\in E(ij)}\left(m_{ki,ij}^{l}\right) \tag{8}$$

$$h_{ij}^{l} = \sum\nolimits_{ki\in E(ij)} \alpha_{ki,ij}^{l} * f_{update}(h_{ki}^{l-1}, h_{i}^{l-1}) \tag{9}$$

$$h_{ij}^{l} = \frac{1}{N}\sum_{n=1}^{N}\sum\nolimits_{ki\in E(ij)} \alpha_{ki,ij}^{l,n} * f_{update}^{n}(h_{ki}^{l-1,n}, h_{i}^{l-1,n}) \tag{10}$$

Multi-head attention network

We built our model on the graph neural network that extracts the features focusing on the nodes and edges, typically capturing local structural features. To incorporate local and global crucial features, we adopted a multi-head attention layer known for capturing distant nodes. Before applying this method, we merged the hidden states from the node and edge sub-models. As shown in Eq. 11, we aggregated the edges connected to the *i*th node and the node h_i^L in the last (*L*th) layer of the sub-models, obtaining a new initial hidden state h_i^{t0} using the function f_{update} (Fig. 1D).

Once the new hidden states containing the local structural features were initialized, we applied the multi-head attention method to our model. The technique allowed us to determine the contributions of these local features by using the *softmax* function to calculate the importance of each sub-structure. In Eq. 12, the attention score $\alpha_{i,j}^{t}$ representing the importance of the *j*th node relative to the *i*th node, is derived using the *softmax* function on the hidden states of nodes h_i^{t-1} and h_j^{t-1}, where t indicates the layer depth. The hidden state h_i^t is then systematically calculated by summing all the nodes in the graph (*N(G)*) with their attention scores (Eq. 13). Finally, we applied the multi-head attention mechanism with k heads to our model (Eq. 14).

$$h_i^{t0} = f_{update}(h_i^L, \sum_{j\in N(i)} h_{ji}^L) \tag{11}$$

$$\alpha_{i,j}^{t} = Softmax_{j\in N(G)}\left(f_{query}(h_i^{t-1}), f_{key}(h_j^{t-1})^T\right) \tag{12}$$

$$h_i^t = \sum\nolimits_{j \in N(G)} \alpha_{i,j}^t * f_{value}(h_j^{t-1}) \tag{13}$$

$$h_i^t = \frac{1}{K} \sum_{k=1}^{K} \sum\nolimits_{j \in N(G)} \alpha_{i,j}^{t,k} * f_{update}^k(h_j^{t-1,k}) \tag{14}$$

Readout phase

Following the multi-head attention layers, we adopted the readout method in our model to extract the hidden feature H of the molecule. For all nodes in the graph ($N(G)$), h_i^T from the last ($= T$th) multi-head attention layer was averaged (Eq. 15). Finally, the feed-forward neural network was applied to the output of the readout phase, yielding the predicted values.

$$H = \frac{1}{|N(G)|} \sum\nolimits_{i \in N(G)} h_i^T \tag{15}$$

Multi-task learning and ensemble method

We employed binary cross-entropy (BCE) and mean squared error (MSE) loss functions for classification and regression tasks, respectively. For classification tasks, we adopted a multi-task learning approach, training a model on multiple tasks simultaneously [4]. This approach leverages larger datasets combined from multiple tasks, which is advantageous when individual tasks have limited data. Multi-task learning also acts as a regularization effect, reducing the risk of overfitting to any specific task. For instance, on the Tox21 dataset, consisting of 12 tasks in NR and SR, multi-task learning yielded better performance than single-task learning. We further enhanced our model through bagging, an ensemble method that improves generalization and mitigates overfitting. In bagging, multiple training datasets are created by sampling subsets with varying distributions, addressing data limitations and providing a diverse training base. Independent models are then trained on these samples, and the final prediction is made by combining the outputs from all trained models. For this ensemble inference, we used soft voting to calculate the final predictions. Unlike hard voting, which selects the prediction favored by the majority of models, soft voting averages the outputs from each model to produce the final prediction. As shown in Eq. 16, the result of sample i is derived by averaging the outputs across multiple models M_k.

$$\widehat{y}_i^t = \frac{1}{K} \sum\nolimits_{k=1}^{K} M_k(x_i^t) \tag{16}$$

Training and evaluation data

Similar to previous studies, we employed the *balanced scaffold splitting method* for evaluation, which introduces a more rigorous and scientifically meaningful challenge

compared to random or scaffold splitting [35, 50]. The *balanced scaffold splitting method* divides molecules into disjoint scaffold sets based on their structural scaffolds. These scaffold sets are then categorized into large and small sets according to a specified size threshold (e.g., half of the size of a test dataset). To ensure the representativeness of the training set and reduce bias in the validation and test datasets, large scaffold sets are consistently allocated to the training dataset.

To train and evaluate our ensemble model, we initially split a dataset using balanced scaffold splitting with a random seed, allocating 10% for testing and 90% for subsequent processing. We then applied a new random seed, referred to as the *e-seed*, distinct from the existing seed, to split the remaining data. Once the *e-seed* was determined, we further divided this data into 80% for training and 10% for validation. This approach enabled us to obtain multiple training-validation dataset pairs using different *e-seed* values, each pair containing slightly different scaffolds in the training dataset. This variability enabled us to train multiple models independently, supporting our ensemble model.

Model training

We trained our ensemble model using datasets created with balanced scaffold splitting. Before training, graphs and line graphs were extracted from the molecules in the datasets using RDKit [32]. Our model was implemented using Pytorch [1], PyTorch Lightning [11], PyTorch Geometric [17], and TorchVision [41]. PyTorch Geometric and TorchVision were specifically used to implement the graph and multi-head attention layers.

We trained our model by minimizing BCE and MSE loss functions for classification and regression tasks, respectively, on the preprocessed datasets. To prevent overfitting and improve model efficiency, we applied an early stopping based on the minimum validation loss [33, 46]. We trained our model on a single NVIDIA Tesla V100-DGXS-32 GB GPU and a CUDA 11.3 environment.

After training, we obtained multiple models from different *e-seed* splits, allowing us to make predictions on the test dataset using an ensemble approach with soft voting. In this method, the final prediction for the test set was obtained by averaging the outputs from each model. Final evaluation metrics, including AUROC for classification tasks and RMSE for regression tasks, were computed using Scikit-Learn [10].

During hyperparameter tuning, we applied early stopping method to prevent overfitting, with a maximum epoch of 1000 and patience of 30 epochs. Patience represents the number of epochs that training continues after a new minimum loss value is found. We optimized our model on validation datasets with various hyperparameters: batch size, layer size, layer depth, learning rate, weight decay, and dropout rate. Detailed hyperparameter configurations are provided in Supplementary Table 1.

We experimented with different batch sizes to assess their effects on overfitting, underfitting, and generalization. The optimal batch sizes were found to be 64 for smaller datasets (e.g., BBBP) and 256 for larger datasets (e.g., ToxCast). We also tested different sizes and depths for the graph and attention layers, which significantly affected model performance. The best performance was achieved with a graph layer depth of three, an attention layer depth of one, and a layer size of 128 for both layers.

For optimization, we adopted the Adam optimizer, known for its stability and speed. We adjusted two key parameters–learning rate and weight decay. Weight decay, similar to L2 regularization, was effective in preventing overfitting. The optimal model configuration was found with a learning rate of 1e-3 and a weight decay of zero. Additionally, dropout was applied to further mitigate overfitting, with the highest performance observed at a dropout rate of 0.3.

Methods for performance comparison

We conducted a comprehensive comparison of our model with various existing methods of different approaches, including graph-based [19, 29, 31, 34, 39, 40, 51, 62, 63], SMILES-based [58], and pre-training methods [16, 22, 25, 35, 50, 58], using the MoleculeNet dataset. This thorough evaluation ensured the fairness and reliability of our model.

Firstly, we compared our model with nine graph-based methods built on the message-passing neural network framework, including AttentiveFP [62], DMPNN [63], TrimNet [34], and CD-MVGNN [40]. AttentiveFP integrates a small graph into a large one using a graph attention mechanism to extract the critical substructures. DMPNN treats bonds as nodes to solve the *tottering* problem, while TrimNet exploits both bond and atom features to enhance the representation of molecular features. CD-MVGNN integrates atom- and bond-based GNNs with *disagreement* loss, mitigating the difference in predictions between these GNNs.

Secondly, we evaluated our model against the SMILES-BERT model [58], which uses the BERT architecture with SMILES string as input to represent chemical structures.

Lastly, we compared our model with six pre-training methods, such as GROVER [50], MPG [35], and KANO [16]. These methods were pre-trained on a large datasets like ZINC [24] and fine-tuned on small datasets like MoleculeNet. GROVER employs multi-level pre-tasks predicting masked information and motifs, MPG used self-supervised learning with the tasks of predicting the masked atom information and pairwise half-graph discrimination. KANO introduces contrastive learning between the augmented molecular graph for the knowledge graph of elements and the enhanced molecular graph for functional groups.

Results

We evaluated the accuracy of our model by comparing it to state-of-the-art models. To assess the stability and generalization of the models, we tested our model and several models across different data characteristics using various random seed values. Additionally, we analyzed the effectiveness of our ensemble method in handling data scarcity. An ablation study provided insight into the contribution of each model component, offering a comprehensive understanding of their roles and impact.

Performance evaluation on MoleculeNet dataset

We evaluated our model with eight state-of-the-art graph-based models and six pre-trained models [16, 19, 22, 25, 29, 31, 34, 35, 39, 50, 51, 58, 62, 63]. These methods were tested on two datasets: *dataset-a*, initially used to validate graph-based methods [50], and *dataset-b,* initially used to evaluate pre-trained models [35]. Both datasets were split using different random seeds for scaffold splitting, resulting in varying

scaffold distributions across the training, validation, and test sets. All models, including ours, were evaluated on consistent test datasets to ensure fair comparison. We measured AUROC for classification tasks and RMSE for regression tasks, averaging results over three random seed values for comparison.

On *dataset-a,* initially used in GROVER [50], our model achieved an average AUROC of 0.821 across six classification tasks, placing second highest. GROVER led with an average AUROC of 0.834, followed by KANO with 0.814 as the third (Table 3). For regression tasks, GROVER achieved the lowest average RMSE scores of 1.544 and 0.560 on FreeSolv and Lipo, respectively, while our method reached the lowest RMSE of 0.746 on ESOL. On *dataset-b,* initially used in MPG [35], our model showed strong performance with an average AUROC of 0.851 on six classification tasks, followed by MPG and the KANO with AUROCs of 0.841 and 0.833, respectively. In regression tasks on FreeSolv, ESOL, and Lipo datasets, MPG, KANO, and our method achieved the lowest RMSE scores of 1.269, 0.405, and 0.545, respectively (Table 4).

Table 3 Performance comparison with current state-of-the-art methods on *dataset-a*

Model	Dataset									Data
	Classification (AUROC)						Regression (RMSE)			
	BBBP	Tox21	ToxCast	SIDER	ClinTox	BACE	FreeSolv	ESOL	Lipo	
MGCN	0.850	0.707	0.663	0.552	0.634	0.734	3.349	1.266	1.113	
[39]	(0.064)	(0.016)	(0.009)	(0.018)	(0.042)	(0.030)	(0.097)	(0.147)	(0.041)	
SchNet	0.847	0.767	0.679	0.545	0.717	0.750	3.215	1.045	0.909	
[51]	(0.024)	(0.025)	(0.021)	(0.038)	(0.042)	(0.033)	(0.755)	(0.064)	(0.098)	
Weave	0.837	0.741	0.678	0.543	0.823	0.791	2.398	1.158	0.813	
[29]	(0.065)	(0.044)	(0.024)	(0.034)	(0.023)	(0.008)	(0.250)	(0.055)	(0.042)	
GraphConv	0.877	0.772	0.650	0.593	0.845	0.854	2.900	1.068	0.712	
[31]	(0.036)	(0.041)	(0.025)	(0.035)	(0.051)	(0.011)	(0.135)	(0.050)	(0.049)	
HU.et.al	0.915	0.811	0.714	0.614	0.762	0.851	-	-	-	
[22]	(0.040)	(0.015)	(0.019)	(0.006)	(0.058)	(0.027)				
AttentiveFP	0.908	0.807	0.579	0.605	0.933	0.863	2.030	0.853	0.650	
[62]	(0.050)	(0.020)	(0.001)	(0.060)	(0.020)	(0.015)	(0.420)	(0.060)	(0.030)	
MPNN	0.913	0.808	0.691	0.595	0.879	0.815	2.185	1.167	0.672	[a][a]
[19]	(0.041)	(0.024)	(0.013)	(0.030)	(0.054)	(0.044)	(0.952)	(0.430)	(0.051)	
DMPNN	0.919	0.826	0.718	0.632	0.897	0.852	2.177	0.980	0.653	
[63]	(0.030)	(0.023)	(0.011)	(0.023)	(0.040)	(0.053)	(0.914)	(0.258)	(0.046)	
MPG	0.935	0.805	0.712	0.628	0.915	0.839	2.286	0.908	0.637	
[35]	(0.012)	(0.015)	(0.010)	(0.020)	(0.023)	(0.052)	(0.389)	(0.105)	(0.044)	
GROVER	0.940	**0.831**	0.737	**0.658**	**0.944**	**0.894**	**1.544**	0.831	**0.560**	
[50]	(0.019)	**(0.025)**	(0.010)	**(0.023)**	**(0.021)**	**(0.028)**	**(0.397)**	(0.120)	**(0.035)**	
KANO	0.939	0.808	0.723	0.626	0.938	0.852	1.938	0.840	0.596	
[16]	(0.016)	(0.018)	(0.008)	(0.009)	(0.022)	(0.034)	(0.288)	(0.150)	(0.036)	
Our model	**0.956**	0.822	**0.749**	0.626	0.921	0.849	1.940	**0.746**	0.597	
	(0.018)	(0.029)	**(0.001)**	(0.007)	(0.035)	(0.028)	(0.312)	**(0.093)**	(0.027)	

Average AUROC and RMSE scores used three random seed values

The numbers in parenthesis are standard deviations of the results on three random seed values

Bold means pretraining model

The bold number and underlined number mean the highest and second-highest numbers

[a] [a] is from [50]

Interestingly, several models showed substantial performance differences between the two datasets. Our method and KANO demonstrated stable performance across both datasets, whereas GROVER and MPG showed significant performance variations. These inconsistencies may be due to structural differences in the scaffold distributions of the two datasets, suggesting that each model focused on different molecular regions during training. To address this and ensure fair comparisons, we conducted an extensive evaluation using 30 random seed values, detailed in Sect. "Performance evaluation with multiple random seed values".

Performance evaluation with multiple random seed values

The performance variations observed between *dataset-a* and *dataset-b* for several methods (Sect. "Performance evaluation on MoleculeNet dataset") underscore the need to assess robustness across different dataset compositions. Thus, using 30 random seed values, we compared our model to prominent methods, including GROVER, MPG, CD-MVGNN, and KANO, on nine benchmark datasets. GROVER [50] and MPG [35] were selected due to their notable performance difference, KANO [16] as a recent advancement in the field, and CD-MVGNN [40] for its multi-view approach that considers both atoms and bonds for non-isomorphic graph differentiation.

As shown in Table 5, our method attained higher average AUROC scores of 0.957, 0.740, 0.628, and 0.919 on the BBBP, ToxCast, SIDER, and ClinTox, respectively. In the regression tasks, our method also obtained the lowest average RMSE scores of 0.783 and 0.597 on the ESOL and Lipo datasets. Meanwhile, KANO and CD-MVGNN outperformed other models on Tox21 and BACE datasets, with AUROC scores of 0.837

Table 4 Performance comparison with current state-of-the-art methods on *dataset-b*

Model	Dataset									Data
	Classification (AUROC)						Regression (RMSE)			
	BBBP	Tox21	ToxCast	SIDER	ClinTox	BACE	FreeSolv	ESOL	Lipo	
Mol2Vec	0.876	0.805	0.690	0.601	0.828	0.841	5.752	2.358	1.178	
[25]	(0.030)	(0.015)	(0.014)	(0.023)	(0.023)	(0.052)	(1.245)	(0.452)	(0.054)	
TrimNet	0.892	0.812	0.652	0.606	0.906	0.843	2.529	1.282	0.702	
[34]	(0.025)	(0.019)	(0.032)	(0.006)	(0.017)	(0.025)	(0.111)	(0.029)	(0.008)	
SMILES-BERT	**0.959**	0.803	0.655	0.568	**0.985**	0.849	2.974	0.841	0.666	
[58]	**(0.009)**	(0.010)	(0.010)	(0.031)	**(0.014)**	(0.021)	(0.510)	(0.096)	(0.029)	
MPG	0.922	0.837	0.748	0.658	0.963	0.920	**1.269**	0.741	0.556	[b][a]
[35]	(0.012)	(0.019)	(0.005)	(0.012)	(0.028)	(0.013)	**(0.192)**	(0.017)	(0.017)	
GROVER	0.883	0.794	0.698	0.598	0.935	0.915	1.713	0.595	0.858	
[50]	(0.024)	(0.021)	(0.007)	(0.028)	(0.049)	(0.037)	(0.249)	(0.309)	(0.054)	
KANO	0.947	0.841	0.740	0.646	0.906	0.919	1.311	**0.405**	0.579	
[16]	(0.018)	(0.017)	(0.019)	(0.015)	(0.101)	(0.051)	(0.107)	**(0.047)**	(0.009)	
Our model	0.951	**0.853**	**0.755**	**0.661**	0.958	**0.929**	1.402	0.736	**0.545**	
	(0.027)	**(0.004)**	**(0.011)**	**(0.024)**	(0.040)	**(0.022)**	(0.197)	(0.065)	**(0.032)**	

Average AUROC and RMSE scores used three random seed values

The numbers in parenthesis are standard deviations of the results on three random seed values

Bold means pretraining model

The bold number and underlined number mean the highest and second-highest numbers

[a] [b] is from [35]

and 0.874, respectively. KANO also achieved the best score of 1.970 in the regression task on the FreeSolv dataset. These results demonstrate the robustness of our model in handling data variations, showing the effectiveness of the ensemble method, particularly for small datasets. Statistical significance was validated through Student's t-test and Welch's t-test (Supplementary Fig. 1 and 2).

Among the five models evaluated on nine datasets, our model ranked the best on six datasets and the second-best on two others. These highlight the importance of developing more generalized and integrated models. Additionally, CD-MVGNN displayed the best performances on one dataset and the second-best performances on the two. These results suggest that multi-view approaches consistently improved the prediction of molecular properties. KANO presented the best scores on two datasets and the second-best scores on the three, demonstrating the effectiveness of the pretraining and knowledge- guided learning. In this context, the balanced performance of our model across multiple datasets highlights its potential for a wide range of molecular property prediction tasks, and our method could promise better predictions with pretraining in future work.

Contribution of the ensemble model

We applied an ensemble method to our model to address data scarcity on the nine MoleculeNet datasets, which contain only a few thousand samples. To validate the effectiveness of the ensemble method, we compared our ensemble model to single models across 30 random seed values. We used balanced scaffold splitting to create six single models with distinct *e-seed* values (from 0 to 5), training each model on a unique training and validation dataset (Figs. 2 and 3). The ensemble model was then constructed by aggregating these six models using soft voting.

Table 5 Performance comparison with sufficient random seed values

	Dataset								
	Classification (AUROC)						**Regression (RMSE)**		
	BBBP	**Tox21**	**ToxCast**	**SIDER**	**ClinTox**	**BACE**	**FreeSolv**	**ESOL**	**Lipo**
GROVER	0.900	0.787	0.664	0.599	0.854	0.815	2.376	0.983	0.805
(Rong, et al., 2020)[50]	(0.034)	(0.041)	(0.029)	(0.036)	(0.142)	(0.046)	(0.891)	(0.175)	(0.072)
MPG	0.919	0.817	0.713	<u>0.625</u>	0.908	<u>0.872</u>	2.165	0.878	0.637
[35]	(0.025)	(0.021)	(0.016)	(0.024)	(0.031)	(0.044)	(0.594)	(0.128)	(0.043)
CD-MVGNN	0.907	**0.837**	<u>0.729</u>	0.623	0.885	0.870	2.049	0.825	<u>0.600</u>
(Ma, et al., 2022) [40]	(0.032)	(0.020)	(0.015)	(0.030)	(0.047)	(0.048)	(0.613)	(0.097)	(0.055)
KANO	<u>0.951</u>	0.820	0.719	0.614	<u>0.910</u>	**0.874**	**1.970**	<u>0.798</u>	0.618
(Fang, et al., 2023) [16]	(0.016)	(0.023)	(0.012)	(0.026)	(0.040)	(0.043)	(0.583)	(0.091)	(0.046)
Our	**0.957**	<u>0.836</u>	**0.740**	**0.628**	**0.919**	0.854	<u>2.019</u>	**0.783**	**0.597**
	(0.017)	(0.021)	(0.014)	(0.027)	(0.037)	(0.045)	(0.532)	(0.074)	(0.038)

Average AUROC and RMSE scores used 30 random seed values

Bold means pretraining model

The bold number and underlined number mean the highest and second-highest numbers

Our results show that the ensemble model consistently outperformed each individual model. For classification tasks, the ensemble model achieved higher AUROC scores than all single models (Fig. 2). The ensemble model improved AUROC scores by 1.4%, 2.4%, 3.0%, 3.3%, 2.9%, and 3.4% on average for the BBBP, Tox21, ToxCast, SIDER, ClinTox, and BACE, respectively. For regression, the ensemble model achieved RMSE improvements of 13.8%, 10.3%, and 8.3% on average for the FreeSolv, ESOL, and Lipo datasets, respectively (Fig. 3). These results indicate that the ensemble method significantly enhances performance, especially in regression tasks. Detailed classification and regression results are presented in Supplementary Table 3 and 4.

Ablation study

We conducted a systematic ablation study to assess the impact of individual components within our model, using MoleculeNet physiology datasets from *dataset-b*. We first evaluated the complete model (M1 in Fig. 4), which integrates all components such as sub-model for node, sub-model for edge, multi-head attention, and ensemble method. We then tested a variant of the model without the ensemble method (M2 in Fig. 4). From the M2 model, we built and evaluated three additional models, each independently eliminating one component – either the sub-model for node, sub-model for edge, and multi-head attention.

The ablation study revealed that each component significantly contributed to the overall performance, with the full model achieving the best performance, underscoring the importance of incorporating all components (Fig. 4). The basic models containing only a single component (either the atom graph, bond graph, or attention) showed lower AUROC scores of 0.795, 0.801, and 0.777, respectively, which are approximately 5%, 4%, and 7% lower than our full model. Notably, the comparison between the M1 and M2 models showed a significant 2.3% increase in AUROC score attributed to the ensemble method. Additionally, we observed AUROC improvements ranging from 1.0% to 5.1% when comparing the M2 model to the other models lacking specific components. Detailed results on the physiology datasets are presented in Supplementary Table 5.

Discussion

As with many clinical prediction tasks, this study faced limitations due to the availability of experimentally labeled data, resulting in small sample sizes that can challenge the training of deep learning models and potentially compromise robustness and generalization. Despite these challenges, our multi-view model, which integrates atom, bond, and global features alongside an ensemble method, showed consistently stable performance across most experiments. These findings suggest that our approach to enhancing generalization could be valuable for clinical applications where data availability is limited. We believe this model holds promise for clinical studies, potentially facilitating reliable predictions in scenarios with constrained data resources.

In addition, the ensemble method has a limitation, increasing computational cost as the number of models increases. Based on our experiments, our results indicate that while our model uses higher computational resources than other architectures, it still performs efficiently within practical limits. For example, we measured training time for MPG [35] and our method; MPG is one of the pretrained methods, and our method

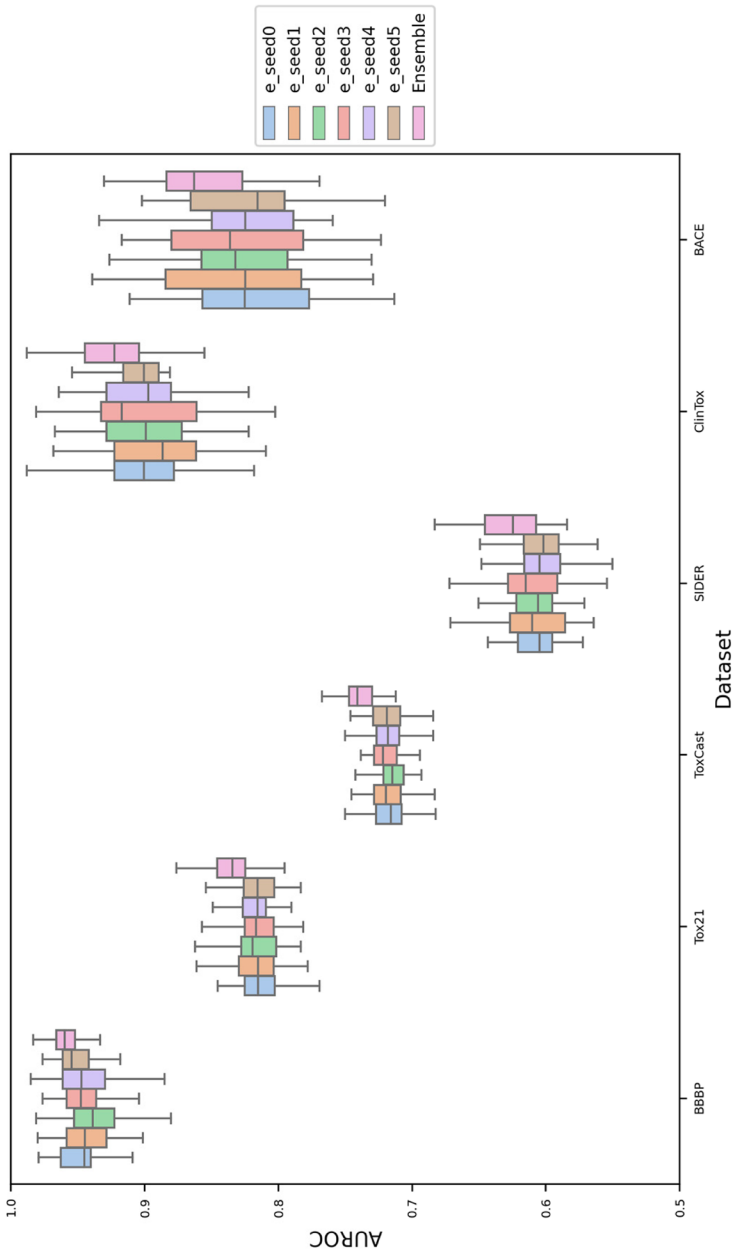

Fig. 2 Performance comparison of the ensemble and individual models on classification tasks. We evaluated our ensemble and individual models across 30 random seeds on the classification datasets to validate the robustness and generalization of the ensemble method. Each *e-seed n* model was independently trained on different training and validation datasets splitting with different *e-seed* values. The *Ensemble* model used soft voting over the six *e-seed n* models. The test dataset was fixed, and all models were evaluated on the same test dataset on a specific seed value. Notable variations in AUROC values were observed among the *e-seed n* models. The *Ensemble* model consistently achieved the best performances across all datasets. Significant improvements of 0.009, 0.018, 0.020, 0.016, 0.022, and 0.022 AUROC over the second-highest scores were observed in BBBP, Tox21, ToxCast, SIDER, ClinTox, and BACE, respectively

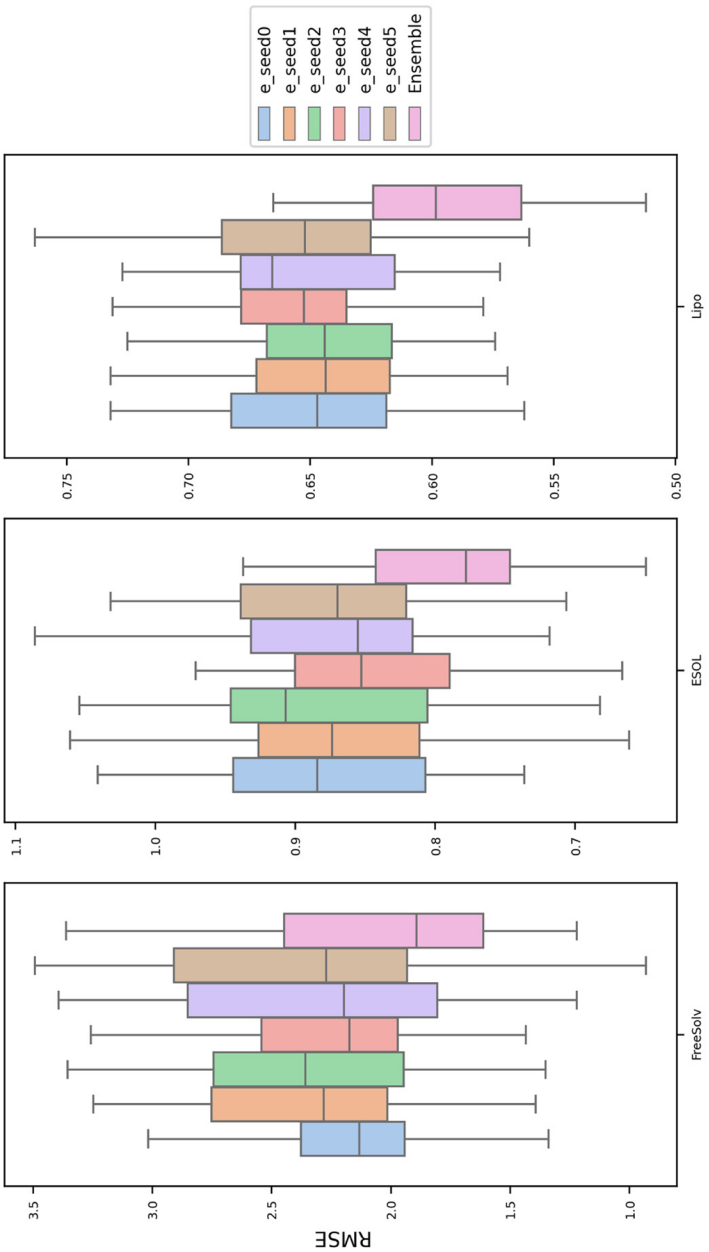

Fig. 3 Performance comparison of the ensemble and individual models on regression tasks. We evaluated our ensemble and individual models across 30 random seeds on the regression datasets. Each *e-seed n* model was independently trained, and the *Ensemble* model used soft voting over the six *e-seed n* models. The *Ensemble* model consistently outperformed across all datasets. Significant improvements of 0.202, 0.060, and 0.047 RMSE over the second-lowest scores were observed in FreeSolv, ESOL, and Lipo, respectively

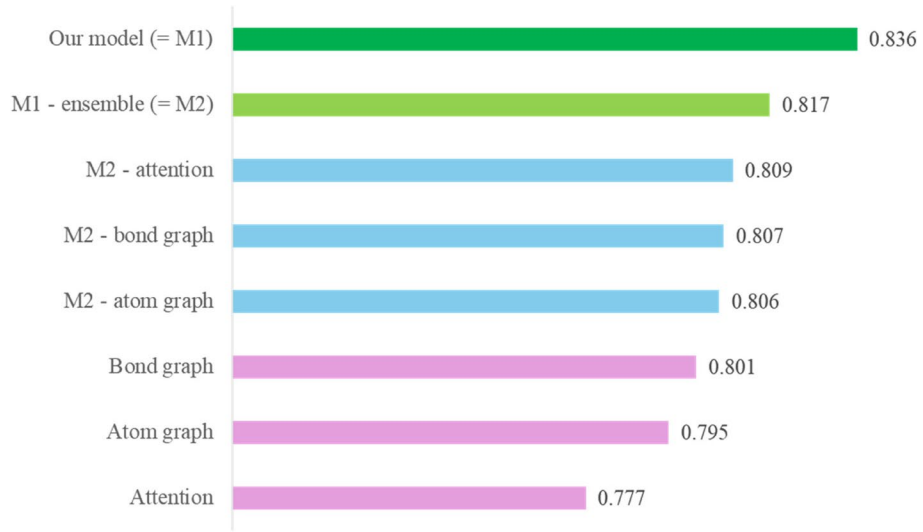

Fig. 4 Ablation study evaluating the contribution of four components in our model. M1 (green) represents the full model, incorporating the atom graph, bond graph, attention mechanism, and ensemble method. M2 (light green) includes all components except the ensemble method. The next three models (light blue) each exclude one component-either the attention mechanism, bond graph, or atom graph. The last three models (pink) include only one component each (either the attention mechanism, bond graph, or atom graph). The full model, M1, achieved the highest performance, with an average AUROC score of 0.836 on MoleculeNet physiology datasets

exploits the ensemble method. To compare these two methods, we chose the BBBP dataset, which includes a small amount of data and one task, and the ToxCast dataset, which contains a relatively large amount of data and 617 tasks. On the BBBP dataset, MPG takes 27 min for finetuning, and our method takes 24 min for the ensemble. If a dataset is small, the ensemble method works efficiently through optimization, such as early stopping. For the ToxCast dataset, MPG takes 46 min, and our method takes one hour and 21 min. Following our anticipation, we observed that a large dataset could demand higher computational resources for the ensemble method, but it still has been within practical limits. Additionally, we can apply several potential optimizations to reduce computational costs, such as reducing the number of attention layers without significant loss of accuracy and leveraging sparse attention mechanisms that selectively focus on critical substructures when it was applied to larger datasets or more complex molecular structures.

To address these challenges associated with the ensemble method, future work should be focused on developing more efficient and practical ensemble approaches for deep learning models. Pretraining methods might be synergetic to the ensemble method by diminishing learning time with fine-tuning. Considering KANO achieved better performances in some of our experiments, adopting the ensemble method to a pre-trained model could promise to reduce computational costs and enhance performance. Additionally, replacing soft voting with attention-based networks could provide improved interpretability, which is essential for many clinical applications.

Conclusions

This study presents a graph-integrated model designed to enhance the prediction of molecular properties by effectively capturing both local and global substructural features. The model incorporates graph attention encoders and multi-head attention mechanisms to a deeper structural understanding. The graph attention networks enable our model to detect more specific structural properties by considering interactions between atoms and edges, while the multi-head attention network captures the relationships between both proximate and distant substructures. This constitution could lead our model to reflect locally and globally essential substructures. Furthermore, the multi-task learning method was adapted to increase performance by facilitating task cooperation and employed an ensemble method to enhance robustness and generalization.

We validated the effectiveness of our model through experiments across 30 random seed values on nine MoleculeNet datasets, comparing it with several state-of-the-art methods. Our model outperformed other models on four classification and two regression datasets, where the results demonstrate higher predictive accuracy. These results underscore the advantage of our ensemble method and highlight the importance of integrating local and global structural information in predicting molecular properties.

Abbreviations

AUROC	Area Under the Receiver Operating Characteristic curve
RMSE	Root Mean Squared Error
QSAR	Quantitative Structure–Activity Relationship
ECFP	Extended-Connectivity FingerPrints
GNN	Graph Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
MPNN	Message-Passing Neural Network
DMPNN	Directional Message-Passing Neural Network
SMILES	Simplified Molecular Input Line Entry System
Lipo	Lipophilicity
ELU	Exponential Linear Unit
BCE	Binary Cross-Entropy
MSE	Mean Squared Error

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13040-024-00419-4.

Supplementary Material 1.

Authors' contributions

MR and HM performed conceptualization and developed the model. HM implemented the algorithm. MR and HM evaluated the model and wrote the manuscript.

Funding

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) [No. 2020–0-01373, Artificial Intelligence Graduate School Program (Hanyang University)] and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023–00217123).

Data availability

No datasets were generated or analysed during the current study.

Declarations

Competing interests

The authors declare no competing interests.

Received: 20 September 2024 Accepted: 26 December 2024
Published online: 16 January 2025

References

1. Ansel J, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In, Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. 2024;2:929–947.
2. Breiman L. Random forests Machine learning. 2001;45:5–32.
3. Brody S, Alon U, Yahav E. How Attentive are Graph Attention Networks? In, *International Conference on Learning Representations*. 2022.
4. Caruana R. Multitask learning Machine learning. 1997;28:41–75.
5. Choudhary K, DeCost B. Atomistic line graph neural network for improved materials property predictions. npj Comput Mat. 2021;7(1):185.
6. Coley CW, et al. Convolutional embedding of attributed molecular graphs for physical property prediction. J Chem Inf Model. 2017;57(8):1757–72.
7. Cortes C, Vapnik V. Support-vector networks Machine learning. 1995;20:273–97.
8. Durant JL, et al. Reoptimization of MDL keys for use in drug discovery. J Chem Inf Comput Sci. 2002;42(6):1273–80.
9. Duvenaud DK, *et* al. Convolutional networks on graphs for learning molecular fingerprints. Advances in neural information processing systems. 2015;28.
10. Fabian P. Scikit-learn: Machine learning in Python. J Machine Learn Res. 2011;12:2825.
11. Falcon W, et al. PyTorchLightning/pytorch-lightning: 0.7. 6 release. Zenodo; 2020.
12. Fan X-Q, et al. I-DNAN6mA: Accurate Identification of DNA N6-Methyladenine Sites Using the Base-Pairing Map and Deep Learning. J Chem Inf Model. 2023;63(3):1076–86.
13. Fan XQ., et al. Ense-i6mA: Identification of DNA N6-methyl-adenine Sites Using XGB-RFE Feature Se-lection and Ensemble Machine Learning. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2024.
14. Fan X, Lin B, Guo Z. Infrared Polarization-Empowered Full-Time Road Detection via Lightweight Multi-Pathway Collaborative 2D/3D Convolutional Networks. 2024. IEEE Transactions on Intelligent Transportation Systems.
15. Fang X, et al. Geometry-enhanced molecular representation learning for property prediction. Nature Machine Intelligence. 2022;4(2):127–34.
16. Fang Y, et al. Knowledge graph-enhanced molecular contrastive learning with functional prompt. Nature Machine Intelligence. 2023;5(5):542–53.
17. Fey M, Lenssen JE. Fast graph representation learning with PyTorch Geometric. arXiv preprint. arXiv:1903.02428. 2019
18. Gasteiger J, Groß J, Günnemann S. Directional Message Passing for Molecular Graphs. In, *International Conference on Learning Representations*. 2020.
19. Gilmer J, et al. Neural message passing for quantum chemistry. In, International conference on machine learning. PMLR; 2017;1263–1272.
20. Gómez-Bombarelli R, et al. Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent Sci. 2018;4(2):268–76.
21. Hansch C, Fujita T. p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. J Amer Chem Soc. 1964;86(8):1616–26.
22. Hu* W, et al. Strategies for Pre-training Graph Neural Networks. In, International Conference on Learning Representations. 2020.
23. Huang R, et al. Tox21Challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. Front Environ Sci. 2016;3:85.
24. Irwin JJ, et al. ZINC20—a free ultralarge-scale chemical database for ligand discovery. J Chem Inf Model. 2020;60(12):6065–73.
25. Jaeger S, Fulle S, Turk S. Mol2vec: unsupervised machine learning approach with chemical intuition. J Chem Inf Model. 2018;58(1):27–35.
26. Jiang D, et al. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. Journal of cheminformatics. 2021;13:1–23.
27. Jiang M, et al. Drug–target affinity prediction using graph neural network and contact maps. RSC Adv. 2020;10(35):20701–12.
28. Karimi M, et al. DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. Bioinformatics. 2019;35(18):3329–38.
29. Kearnes S, et al. Molecular graph convolutions: moving beyond fingerprints. J Comput Aided Mol Des. 2016;30:595–608.
30. Kim S, et al. PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res. 2021;49(D1):D1388–95.
31. Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks. In, International Conference on Learning Representations. 2017.
32. Landrum G. Rdkit documentation Release. 2013;1(1–79):4.
33. Li M, Soltanolkotabi M, Oymak S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In, International conference on artificial intelligence and statistics. PMLR; 2020. p. 4313–4324.
34. Li P, et al. TrimNet: learning molecular representation from triplet messages for biomedicine. Brief Bioinform. 2021;22(4):bbaa266.
35. Li P, et al. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. Brief Bioinform. 2021;22(6):bbab109.
36. Li S et al. Geomgcl: Geometric graph contrastive learning for molecular property prediction. In, Proceedings of the AAAI conference on artificial intelligence. 2022. p. 4541–4549.

37. Lin X, et al. Interpretable multi-view attention network for drug-drug interaction prediction. In, 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE; 2023. p. 422–427.
38. Liu H, et al. Attention-wise masked graph contrastive learning for predicting molecular property. Brief Bioinform. 2022;23(5):bbac303.
39. Lu C, et al. Molecular property prediction: A multilevel quantum interactions modeling perspective. In, Proceedings of the AAAI conference on artificial intelligence. 2019. p. 1052–1060.
40. Ma H, et al. Cross-dependent graph neural networks for molecular property prediction. Bioinformatics. 2022;38(7):2003–9.
41. Maintainers, T. and Contributors. 2016. TorchVision: PyTorch's Computer Vision library. https://github.com/pytorch/vision
42. Martins IF, et al. A Bayesian approach to in silico blood-brain barrier penetration modeling. J Chem Inf Model. 2012;52(6):1686–97.
43. Mayr A, et al. DeepTox: toxicity prediction using deep learning. Front Environ Sci. 2016;3:80.
44. Nguyen T, et al. GraphDTA: predicting drug–target binding affinity with graph neural networks. Bioinformatics. 2021;37(8):1140–7.
45. Öztürk H, Özgür A, Ozkirimli E. DeepDTA: deep drug–target binding affinity prediction. Bioinformatics. 2018;34(17):i821–9.
46. Prechelt L. Early stopping-but when? In, Neural Networks: Tricks of the trade. In, Neural Networks: Springer; 2002. p. 55–69.
47. Ramsundar B et al. Deep Learning for the Life Sciences. O'Reilly Media.
48. Richard AM, et al. ToxCast chemical landscape: paving the road to 21st century toxicology. Chem Res Toxicol. 2016;29(8):1225–51.
49. Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model. 2010;50(5):742–54.
50. Rong Y, et al. Self-supervised graph transformer on large-scale molecular data. Adv Neural Inf Process Syst. 2020;33:12559–71.
51. Schütt K, et al. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. Adv Neural Inform Process Syst. 2017;30.
52. Somnath VR, et al. Learning graph models for retrosynthesis prediction. Adv Neural Inf Process Syst. 2021;34:9405–15.
53. Steinbeck C, et al. The Chemistry Development Kit (CDK): An open-source Java library for chemo-and bioinformatics. J Chem Inf Comput Sci. 2003;43(2):493–500.
54. Sun R, Dai H, Yu AW. Does gnn pretraining help molecular representation? Adv Neural Inf Process Syst. 2022;35:12096–109.
55. Tang B, et al. A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. Journal of cheminformatics. 2020;12:1–9.
56. Torng W, Altman RB. Graph convolutional neural networks for predicting drug-target interactions. J Chem Inf Model. 2019;59(10):4131–49.
57. Veličković P, et al. Graph Attention Networks. In, International Conference on Learning Representations. 2018.
58. Wang, S., et al. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In, Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics. 2019. p. 429–436.
59. Wang Z, et al. Advanced graph and sequence neural networks for molecular property prediction and drug discovery. Bioinformatics. 2022;38(9):2579–86.
60. Withnall M, et al. Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction. Journal of cheminformatics. 2020;12(1):1.
61. Wu Z, et al. MoleculeNet: a benchmark for molecular machine learning. Chem Sci. 2018;9(2):513–30.
62. Xiong Z, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. J Med Chem. 2019;63(16):8749–60.
63. Yang K, et al. Analyzing learned molecular representations for property prediction. J Chem Inf Model. 2019;59(8):3370–88.
64. Yang N, et al. Learning substructure invariance for out-of-distribution molecular representations. Adv Neural Inf Process Syst. 2022;35:12964–78.
65. Yap CW. PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. J Comput Chem. 2011;32(7):1466–74.
66. Zaslavskiy M, et al. ToxicBlend: Virtual screening of toxic compounds with ensemble predictors. Computational Toxicology. 2019;10:81–8.
67. Zeng Y, et al. Deep drug-target binding affinity prediction with multiple attention blocks. Briefings in bioinformatics. 2021;22(5):bbab117.
68. Zhang Z, Guan J, Zhou S. FraGAT: a fragment-oriented multi-scale graph attention model for molecular property prediction. Bioinformatics. 2021;37(18):2981–7.
69. Zhang Z, et al. Motif-based graph self-supervised learning for molecular property prediction. Adv Neural Inf Process Syst. 2021;34:15870–82.
70. Zhao Q, AttentionDTA: prediction of drug–target binding affinity using attention model. In, et al. IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE. 2019;2019:64–9.

Publisher's Note