Research
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

Molecular Merged Hypergraph Neural Network for Explainable Solvation Gibbs Free Energy Prediction

Wenjie Du[1,2], Shuai Zhang[2], Zhaohui Cai[3,4], Xuqiang Li[1,2], Zhiyuan Liu[5], Junfeng Fang[5*], Jianmin Wang[6*], Xiang Wang[5], and Yang Wang[1,2,7*]

[1]Key Laboratory of Precision and Intelligent Chemistry, University of Science and Technology of China, Hefei, Anhui 230026, China. [2]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, Jiangsu 215123, China. [3]Suzhou Laboratory, Suzhou 215000, China. [4]School of Advanced Technology, Xi'an Jiaotong–Liverpool University, Suzhou 215123, China. [5]School of Computing, National University of Singapore, Singapore 117417, Singapore. [6]Department of Integrative Biotechnology, Yonsei University, Incheon 21983, Republic of Korea. [7]Anhui Provincial Key Laboratory of High Performance Computing, Hefei, China.

*Address correspondence to: jmwang113@hotmail.com (J.W.); fangjf1997@gmail.com (J.F.); angyan@ustc.edu.cn (Y.W.)

Citation: Du W, Zhang S, Cai Z, Li X, Liu Z, Fang J, Wang J, Wang X, Wang Y. Molecular Merged Hypergraph Neural Network for Explainable Solvation Gibbs Free Energy Prediction. *Research* 2025;8:Article 0740. https://doi.org/10.34133/research.0740

Submitted 1 April 2025
Revised 9 May 2025
Accepted 26 May 2025
Published 15 August 2025

Copyright © 2025 Wenjie Du et al. Exclusive licensee Science and Technology Review Publishing House. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License (CC BY 4.0).

Solvation free energies play a fundamental role in various fields of chemistry and biology. Accurately determining the solvation Gibbs free energy (ΔG_{solv}) of a molecule in a given solvent requires a deep understanding of the intrinsic relationships between solute and solvent molecules. While deep learning methods have been developed for ΔG_{solv} prediction, few explicitly model intermolecular interactions between solute and solvent molecules. The molecular modeling graph neural network more closely aligns with real-world chemical processes by explicitly capturing atomic-level interactions, such as hydrogen bonding. It achieves this by initially establishing indiscriminate connections between intermolecular atoms, which are subsequently refined using an attention-based aggregation mechanism tailored to specific solute–solvent pairs. However, its sharply increasing computational complexity limits its scalability and broader applicability. Here, we introduce an improved framework, molecular merged hypergraph neural network (MMHNN), which leverages a predefined subgraph set and replaces subgraphs with supernodes to construct a hypergraph representation. This design effectively mitigates model complexity while preserving key molecular interactions. Furthermore, to handle noninteractive or repulsive atomic interactions, MMHNN incorporates an interpretation mechanism for nodes and edges within the merged graph, leveraging the graph information bottleneck theory to enhance model explainability. Extensive experimental validation demonstrates the efficiency of MMHNN and its improved interpretability in capturing solute–solvent interactions.

Introduction

The interaction properties of solute–solvent in a specific solvent plays a crucial role in physical chemistry, such as chemical reactions [1,2] and electrochemistry [3], among others. A profound understanding of these interactions is essential not only for rationalizing experimental results, but also for guiding the way toward controlling and designing reactions and properties [4,5]. The solvation Gibbs free energy (ΔG_{solv}) is a crucial physicochemical property that determines the behavior of a molecule in solution [6] and is closely linked to the partition coefficient of the solute between the gas and solvent phase [4,7]. In realistic applications, obtaining an accurate ΔG_{solv} result from a large number of solute–solvent combinations can be challenging and expensive using advanced-level methods such as first-principles calculations. Furthermore, changes in solvent and/or solute environment require to be recalculated and reevaluated [1,3,8–10]. Deep learning could serve as an effective tool that has emerged to predict ΔG_{solv} in different solute–solvent systems, providing results quickly without seriously sacrificing accuracy in many cases even when experimental measurements are difficult to obtain.

A wide range of methods have been developed [6,11,12] for ΔG_{solv} prediction, with solute–solvent interaction modeling predominantly relying on 2 key approaches. The first, embedding concatenation, utilizes separate graph neural networks (GNNs) to encode solute and solvent molecules independently. The resulting embeddings are then concatenated for downstream prediction, as illustrated in Fig. 1A. While straightforward, this approach primarily captures intramolecular representations

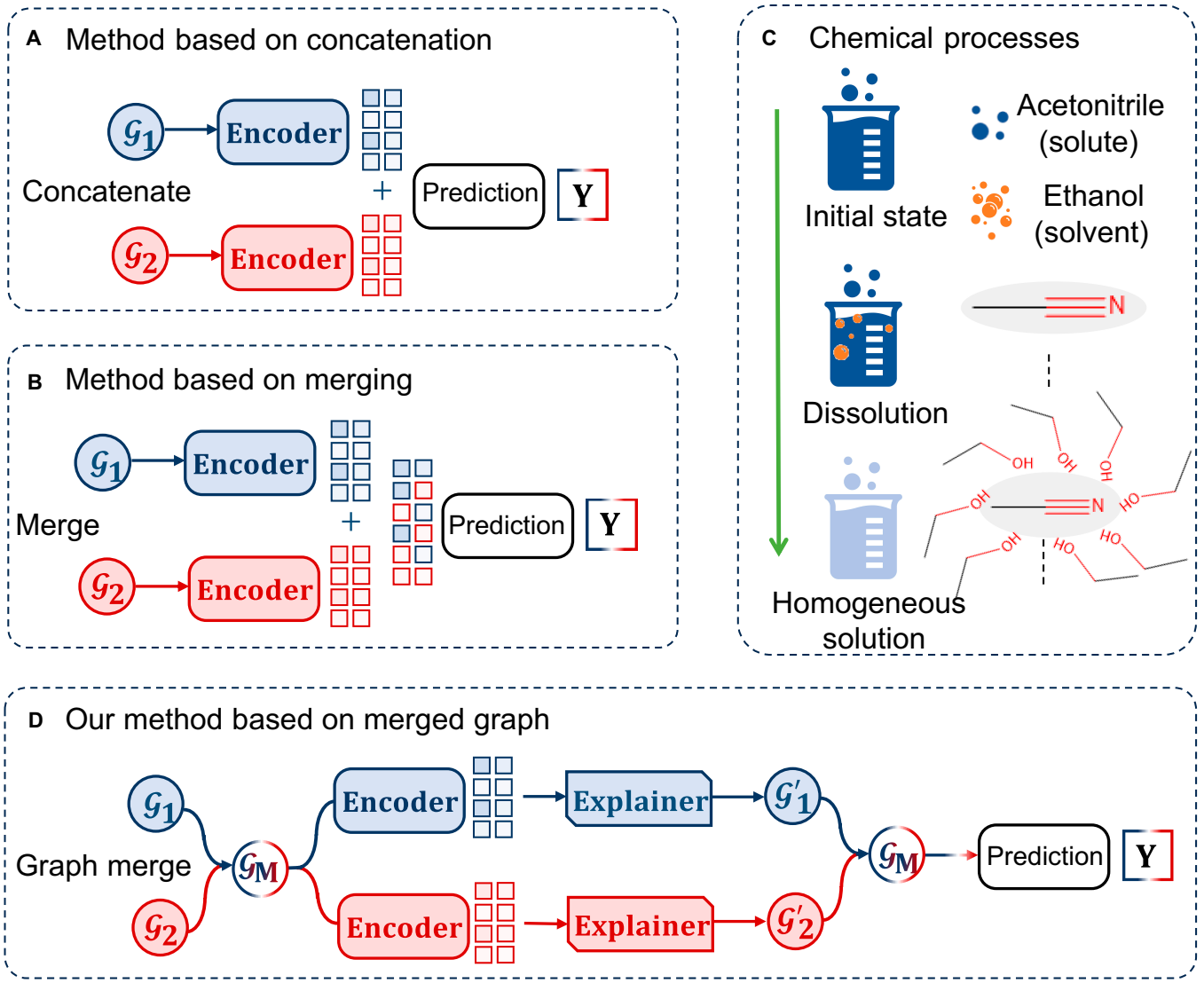


Fig. 1. Illustration of different paradigms for ΔG_{solv} prediction. (A) Embedding concatenation; (B) embedding merging; (C) a schematic diagram of the dissolution process, where acetonitrile is dissolved in ethanol through intermolecular interactions; and (D) our proposed method.

at the node and edge levels within individual molecules, potentially neglecting critical intermolecular interactions that govern solvation behavior. In contrast, the embedding merging, depicted in Fig. 1B, seeks to bridge this gap by explicitly modeling solute–solvent interactions. Like the previous method, it employs 2 graph encoders for initial molecular representation. However, instead of direct concatenation, the embeddings undergo structured integration via advanced interaction mechanisms, such as transformer-based architectures or interactive pruning algorithms, ensuring a more comprehensive reflection of complex intermolecular coupling before prediction. This enhanced representation framework better captures the nuances of solvation free energy, offering a more faithful approximation of molecular interactions in solution.

While the embedding merging approach represents a step forward, it remains insufficient for accurately capturing the underlying chemical processes governing dissolution, as illustrated in Fig. 1C. Dissolution fundamentally involves the replacement of intramolecular forces within the solute and solvent by intermolecular interactions between them. A representative example is the dissolution of acetonitrile in ethanol, where acetonitrile molecules become progressively surrounded by ethanol, forming a homogeneous solution through hydrogen bonding at the atomic (node) level. This natural process underscores the necessity of explicitly modeling intermolecular node interactions, rather than relying solely on embedding-level representations. A model that directly encodes these fine-grained interactions would more faithfully reflect real chemical behavior, enhancing both scientific rigor and predictive accuracy in solvation free energy estimation [13–15].

In order to address to these limitations, we propose the molecular modeling graph neural network (MMGNN) in International Joint Conference on Artificial Intelligence (IJCAI) 2024 [16], which is a novel framework to simulate the actual chemical processes in solute–solvent interactions. MMGNN substantially improves the prediction accuracy of ΔG_{solv} as illustrated in Fig. 1D. Our model has achieved state-of-the-art (SOTA) performance across multiple benchmark datasets, demonstrating superior generalization capability and interpretability [16]. However, we observe 2 critical challenges.

First, the indiscriminate construction of intermolecular edges leads to a rapid increase in computational complexity, particularly for large molecules. Second, treating solute and solvent with the same information propagation strategy may mislead the model, making it unable to distinguish between chemically distinct scenarios, such as dissolving ethanol in water versus dissolving water in ethanol, where the roles of solute and solvent are fundamentally different. Addressing these challenges is essential for further improving both the efficiency and robustness of MMGNN in complex chemical environments.

Hence, we further propose an optimized evaluation method, molecular merged hypergraph neural network (MMHNN), building upon the foundation of MMGNN. Specifically, as depicted within Fig. 2, MMHNN incorporates the following optimizations:

- Predefined subgraph set: The current practice of assigning weights to relationship edges only considers atomic interactions. However, in practical chemical processes such as solvation or binding, spontaneous molecular interactions occur, where van der Waals forces and hydrogen bonding play significant roles. Certain specific atomic and subgraph combinations, such as -OH, -COOH, N, O, F, H, etc. [17–19], exhibit stronger interactions. To address this, we construct a prior knowledge repository to enhance attention mechanisms, allowing for not only the dynamic adjustment of relationship edge weights but also the consideration of interactions between atoms, atom–subgraph, and subgraph–subgraph relationships.
- Merged hypergraph neural network: Constructing relationship edges in pairs, as previously done, is an effective strategy for predicting properties between solute–solvent molecular pairs. However, this approach significantly increases the graph's complexity, especially for large, intricate molecules, due to the substantial rise in degrees of freedom, leading to higher computational resource and time consumption. To address this challenge, we propose a merged hypergraph structure, where a super node aggregates molecular frequency subgraph information using a molecular autoencoder. Information transmission between super nodes facilitates intermolecular message passing, effectively mitigating the issue.
- Explainable method based on GIB: In the merged hypergraph, hypernodes encapsulate critical substructure information. Inspired by graph explainability algorithms, we employ an enhanced information bottleneck theory to derive interpretable results on the hypergraph. MMHNN incorporates interpreters for hypernodes in the merged graph, which aims to reduce redundancy and concentrate on pertinent interactions. The explanatory subgraph is then encoded and used to predict Gibbs free energy with improved accuracy.

A preliminary version of this work was published in IJCAI 2024 [16]. This journal manuscript (MMHNN) extends the initial version in several key aspects. Firstly, MMHNN significantly broadens the application scope and enhances the reliability of MMGNN by introducing a hypergraph construction and a prior knowledge-based reweighting block. Secondly, we extend our investigation to include a variety of solvation free energy prediction tasks, utilizing a range of both molecular interaction and intrinsic explanation methods, aiming for a comprehensive validation of fairness and reliability. Thirdly, considering the computational complexity and time consumption of the model, we evaluated the complexity and runtime of different models, providing a comprehensive assessment of their performance.

Results and Discussion

Overview of the model

The improved novel merged molecular graph model introduces a hypergraph concept where nodes represent substructures, enabling enhanced molecular interactions as shown in Fig. 2. Specifically, a solute–solvent pair is firstly converted into a

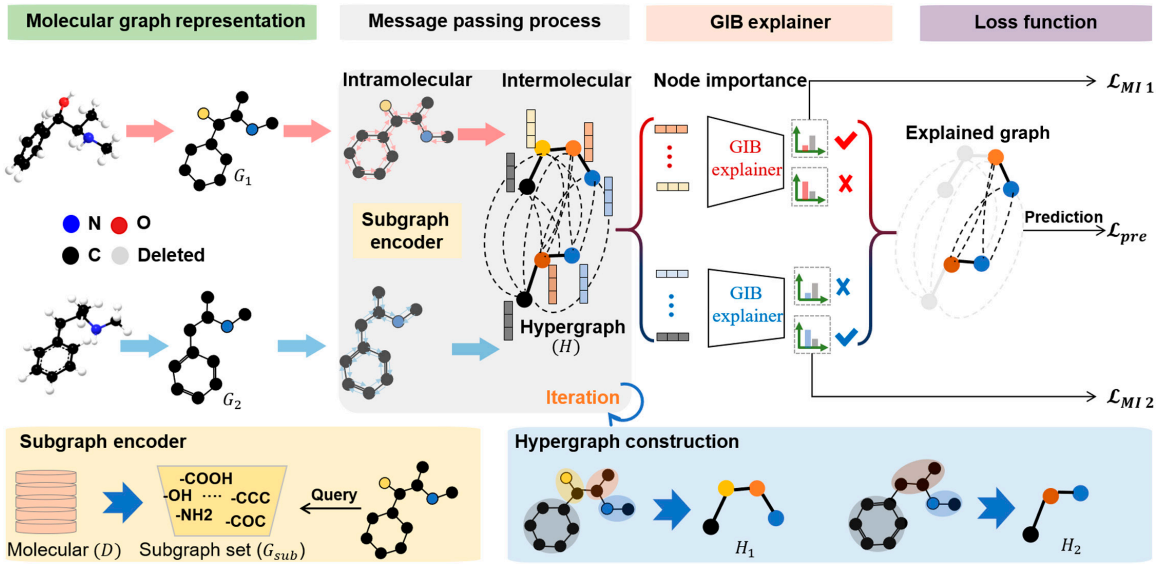


Fig. 2. Overview of the enhanced merged hypergraph neural network model. The model begins by converting an input molecular pair into a topological graph. Substructures are mined and treated as hypernodes, forming a hypergraph to facilitate inter- and intramolecular message passing. The initial layer updates solute atomic features using classical GNN methods, followed by a cross-molecular co-attention mechanism based on hypernodes to refine interactions. Finally, a graph information bottleneck (GIB) principle is applied to highlight significant substructures, improving model interpretability and alignment with biochemical realities. Best viewed in color.

machine-readable topological graph, as detailed in the "Molecular graph representation" section. We then apply frequent subgraph mining to extract substructures, construct a subgraph set, and treat these as hypernodes to form a hypergraph (see the "Hypergraph construction" section). Within this hypergraph, we facilitate both inter- and intramolecular message passing processes. In the initial layer of message passing, each atomic feature of the solute molecule is updated independently using classical GNN methods. Subsequently, a cross-molecular co-attention mechanism based on hypernodes is employed to further update the interaction graph nodes in the "Intramolecular message passing process" and "Intermolecular message passing process" sections. Lastly, to enhance interpretability, we incorporate graph information bottleneck (GIB) principle to capture a core subgraph, as presented in the "Core hypernode extraction based on GIB" section, highlighting the most significant atomic interactions. This augmentation aims to improve the model's generalization capability and better align it with biochemical realities.

A comprehensive overview of the selected atom, bond, and global input features is presented in Table 1. The initial step involves the conversion of the SMILES string of both solute and solvent into a graph structure using the RDKit package [20]. This package is employed not only for graph creation but also for the computation of atom and bond features for each graph. The selection of features was restricted to those computable in RDKit to mitigate the computational expenses associated with performing quantum mechanics calculations for the entire dataset. In order to standardize the lengths of the bond, atom, and global feature vectors, a linear transformation is applied to each vector before the commencement of the message-passing steps.

Experimental setup

Datasets

This study utilizes the CombiSolv-Exp database, compiled by Vermeire and Green [11], which integrates experimental solvation data from multiple sources. Additionally, the FreeSolv database is incorporated, curated by Mobley and Guthrie [21], as well as the CompSol database from Moine et al. [22] and the Abraham dataset, collected by the Abraham group [23]. Furthermore, the Minnesota Solvation (MNSol) Database provides an extensive collection of 3,037 experimentally measured solvation free energies or transfer free energies [24]. These datasets collectively form a comprehensive benchmark for evaluating solvation energy predictions, as presented in Table 2.

Table 1. Atoms (nodes), bonds (edges), and global features for molecular representation

Atomic features ($\mathcal{V}$)	Bond features ($\mathcal{E}$)	Global features ($\mathcal{U}$)
Atomic species	Bond type	Total no. of atoms
No. of bonds	Conjugated status	Total no. of bonds
No. of bonded H atoms	Ring size	Molecular weight
Ring status	Stereo-chemistry	–
Valence	–	–
Aromatic status	–	–
Hybridization type	–	–
Donor status	–	–
Acceptor status	–	–
Partial charge	–	–

Baselines

We compared MMGNN with some SOTA methods, and they could be classified into 2 categories. The first category of the models includes D-MPNN [11], SolvBert [25], SMD [26], Explainable GNN [6], GAT [27], GROVER [28], Uni-Mol [29], and Gem [30] that solely concatenate the molecular representations but ignore the interaction information between solvent and solute. The second category of the models includes methods like CIGIN [31] and CGIB [32], which implicitly considered the intermolecular interactions.

Experimental settings

MMGNN and MMHNN were trained using the Adam optimizer [33] with a learning rate decaying from 1×10^{-4} to 0.5, and a batch size of 50. The mean squared error was used as the training loss. Training was terminated if the validation error did not improve for 150 consecutive epochs or upon reaching the maximum limit of 1,000 epochs. All models were implemented in the PyTorch framework and trained on a Tesla A100 (40 GB) graphics processing unit.

Implementation details

The training is conducted with a batch size of 128. The model consists of a 4-layer GNN for message passing, with both node and edge embedding dimensions set to 300. For graph-level representation, we utilize the set2set pooling strategy. The feedforward neural network is configured with a feedforward dimension of 1,024 and a dropout rate of 0.1. A cosine annealing scheduler is used to adjust the learning rate during training. Early stopping is applied if the validation performance does not improve for 150 consecutive epochs, or training is terminated after a maximum of 1,000 epochs.

Evaluation metrics

Model performance was evaluated using mean absolute error (MAE) and root mean squared error (RMSE). The reported results include the mean and standard deviation, as summarized in the tables.

Table 2. The dataset information

Dataset	Solvent	Solute	Pairs	Reference
FreeSolv	1	643	643	[21]
CompSol	259	442	3,548	[22]
Abraham	122	1,038	6,091	[23]
CompSolv-Exp	296	1,412	10,331	[11]
MNSol	92	790	3,037	[24]

Model performance

Similar to previous studies, a random split is employed to evaluate model performance. All methods were trained, validated, and tested on identical datasets using an 8:1:1 ratio in each test. The experimental results are presented in Table 3. Notably, MMHNN demonstrates superior predictive performance, consistently surpassing other baselines across the 3 test datasets. In terms of the MAE metric, MMHNN achieves an average reduction of 0.012 and 0.32 on the CompSol and CompSol-Exp datasets, respectively, compared to CGIB, the second-best model. Furthermore, performance exhibits a distinct ascending order across different interaction types, with models incorporating molecular interactions outperforming those that do not. This underscores the significance of explicitly modeling intermolecular interactions, as their comprehensive consideration substantially enhances predictive accuracy. The use of molecular merge graphs facilitates a more intrinsic representation of molecular interactions, rendering their relationships more explicit and pronounced.

Further analysis was conducted on the degree of intermolecular information exchange, as presented in Table 4. Various γ values were examined to assess their impact on information transfer. A γ value of 0, which exclusively considers intramolecular interactions, results in the highest prediction error, as indicated by both MAE and RMSE metrics. Increasing the interaction level (γ ranging from 0 to 0.2) significantly improves predictive performance, leading to an 18.6% reduction in MAE. However, further increasing γ beyond 0.2 results in a performance decline, likely due to redundant and excessive information interfering with the model's ability to capture essential patterns. These findings highlight the necessity of balancing intrinsic and intermolecular information to achieve optimal predictive performance.

Additionally, we analyzed the influence of different GNN backbones on molecular encoding strategies. MMHNN is a versatile framework that can be implemented with various GNN architectures, such as GAT, graph convolution network (GCN), and graph isomorphism network (GIN), to extract intermolecular information. In this evaluation, intramolecular interactions were modeled using the optimal γ value of 0.2, as determined in the previous experiments. Notably, the GCN-based model outperformed both GAT and GIN, achieving reductions of approximately 14.8% and 6.01% in MAE, respectively. Consequently, all subsequent experiments adopted the GCN framework for intramolecular message passing and a hyperparameter setting of 0.2 for intermolecular message passing.

Explainability analysis

Four explainability methods, local mask-based, global mask-based, gradient-based, and GIB (detailed descriptions can be found in the Appendix D), are conducted separately in evaluations across various datasets and documented the optimal results for MMHNN. The results are presented in Table 5. Notably, the incorporation of explainability methods leads to a significant improvement in model performance. Among them, the GIB-based method consistently outperforms the other approaches, which means that the introduction of interpretability methods helps the model focus on important relationships, thereby improving model performance.

Further interpretability results are analyzed, and given the current lack of consensus on the key functional groups involved in the energy release process of solution solvents, we compiled

Table 3. Test performance of different methods across 8 independent runs. Mean values are reported, with standard deviations shown in parentheses. (The best result in each column is underlined, while the top-performing baseline is marked with a superscript dagger.)

	MAE (↓)					RMSE (↓)				
Model	FreeSolv	CompSol	Abraham	CompSolv-Exp	MNSol	FreeSolv	CompSol	Abraham	CompSolv-Exp	MNSol
D-MPNN	$0.684_{(0.052)}$	$0.179_{(0.013)}$	$0.454_{(0.036)}$	$0.442_{(0.022)}$	$0.459_{(0.032)}$	$1.164_{(0.055)}$	$0.343_{(0.017)}$	$0.624_{(0.024)}$	$0.672_{(0.051)}$	$0.667_{(0.017)}$
Explainable GNN	$0.724_{(0.031)}$	$0.184_{(0.012)}$	$0.486_{(0.042)}$	$0.321_{(0.013)}$	$0.396_{(0.011)}$	$1.276_{(0.045)}$	$0.367_{(0.012)}$	$0.776_{(0.035)}$	$0.404_{(0.054)}$	$0.673_{(0.024)}$
SolvBERT	$0.588_{(0.034)}$	$0.167_{(0.014)}$	$0.467_{(0.034)}$	$0.382_{(0.023)}$	$0.354_{(0.021)}$	$1.021_{(0.043)}$	$0.328_{(0.020)}$	$0.652_{(0.022)}$	$0.472_{(0.041)}$	$0.623_{(0.104)}$
GAT	$0.675_{(0.033)}$	$0.187_{(0.011)}$	$0.457_{(0.043)}$	$0.970_{(0.031)}$	$0.514_{(0.043)}$	$1.185_{(0.075)}$	$0.390_{(0.012)}$	$0.726_{(0.040)}$	$0.810_{(0.101)}$	$0.812_{(0.124)}$
GROVER	$0.623_{(0.054)}$	$0.155_{(0.022)}^{\dagger}$	$0.307_{(0.035)}$	$0.382_{(0.023)}$	$0.354_{(0.024)}$	$1.015_{(0.022)}$	$0.332_{(0.016)}$	$0.475_{(0.044)}$	$0.491_{(0.053)}$	$0.672_{(0.027)}$
SMD	$0.574_{(0.036)}$	$0.162_{(0.014)}$	$0.374_{(0.024)}$	$0.633_{(0.044)}$	$0.427_{(0.034)}$	$1.113_{(0.015)}$	$0.317_{(0.011)}$	$0.516_{(0.065)}$	$1.023_{(0.152)}$	$0.682_{(0.032)}$
Uni-Mol	$0.565_{(0.038)}$	$0.164_{(0.027)}$	$0.322_{(0.071)}$	$0.214_{(0.022)}$	$0.374_{(0.021)}$	$1.002_{(0.064)}$	$0.303_{(0.020)}$	$0.602_{(0.035)}$	$0.373_{(0.043)}$	$0.657_{(0.019)}$
Gem	$0.584_{(0.041)}$	$0.174_{(0.011)}$	$0.201_{(0.065)}$	$0.253_{(0.023)}$	$0.367_{(0.025)}$	$1.131_{(0.059)}$	$0.290_{(0.019)}$	$0.641_{(0.031)}$	$0.551_{(0.023)}$	$0.675_{(0.027)}$
CIGIN	$0.564_{(0.057)}$	$0.164_{(0.016)}$	$0.254_{(0.010)}$	$0.241_{(0.023)}$	$0.347_{(0.023)}$	$0.910_{(0.015)}$	$0.318_{(0.020)}$	$0.404_{(0.007)}$	$0.411_{(0.032)}$	$0.644_{(0.012)}$
CGIB	$0.531_{(0.034)}^{\dagger}$	$0.156_{(0.014)}$	$0.195_{(0.005)}^{\dagger}$	$0.203_{(0.033)}^{\dagger}$	$0.321_{(0.017)}^{\dagger}$	$0.892(0.022)^{\dagger}$	$0.278_{(0.018)}^{\dagger}$	$0.391_{(0.006)}^{\dagger}$	$0.351_{(0.031)}^{\dagger}$	$0.613_{(0.023)}^{\dagger}$
MMGNN	$0.536_{(0.030)}$	$0.146_{(0.010)}$	$0.187_{(0.008)}$	$0.171(0.013)$	$0.281_{(0.011)}$	$0.902_{(0.026)}$	$0.267_{(0.012)}$	$0.385_{(0.008)}$	$0.303_{(0.033)}$	$0.607_{(0.027)}$
MMHNN	$0.527(0.028)$	$0.144(0.011)$	$0.183_{(0.007)}$	$0.172_{(0.011)}$	$0.271(0.011)$	$0.894_{(0.024)}$	$0.257(0.010)$	$0.379(0.006)$	$0.310(0.027)$	$0.585(0.022)$

Table 4. Test results of the MMHNN under different information interaction update rates and GNN backbones. Interaction coefficient is denoted as Δ. Bold indicates the best results and values are presented as mean (standard deviation).

Information Interaction	Δ and Backbones	MAE (↓)	RMSE (↓)
Intermolecular	$\gamma = 1.0$	$0.192_{(0.023)}$	$0.351_{(0.042)}$
	$\gamma = 0.5$	$0.182_{(0.013)}$	$0.342_{(0.033)}$
	$\gamma = 0.2$	$\mathbf{0.172}_{(0.011)}$	$\mathbf{0.310}_{(0.027)}$
	$\gamma = 0.1$	$0.182_{(0.013)}$	$0.358_{(0.033)}$
	$\gamma = 0.0$	$0.204_{(0.028)}$	$0.384_{(0.055)}$
	GAT	$0.196_{(0.012)}$	$0.383_{(0.045)}$
	GIN	$0.213_{(0.017)}$	$0.442_{(0.035)}$
	GCN	$0.266_{(0.026)}$	$0.452_{(0.048)}$
Intramolecular	GAT	$0.202_{(0.025)}$	$0.412_{(0.037)}$
	GCN	$0.172_{(0.011)}$	$0.310_{(0.027)}$
	GIN	$0.183_{(0.018)}$	$0.352_{(0.026)}$

a set of potentially important functional groups from relevant literature to serve as ground truth for validating algorithmic explainability [34,35]. We visualized pairwise solute–solvent interactions using interaction map weights. Specifically, we examined solvent–solute combinations that induce variations in solvation free energy (ΔG_{solv}) from a chemical perspective by indexing atomic contributions based on attention weights. To enhance interpretability and provide an intuitive representation of functional group interactions, we aggregated the contribution scores for each atom rather than displaying individual atom–pairwise interactions. In the visualizations, solvation-enhancing interactions are highlighted in green, whereas solvation-inhibiting interactions are pink in Fig. 3, with color intensity indicating the relative strength of the effect.

It is evident that the significant substructures identified by MMHNN and MMGNN align closely with the ground-truth annotations, likely due to class-specific patterns captured by the explanatory subnetwork. In contrast, baseline explainability methods occasionally highlight edges that do not correspond to ground-truth annotations. This observation validates MMHNN's capability to capture chemically meaningful substructure interactions, demonstrating its ability to extract implicit scientific knowledge from solute–solvent molecular interactions.

Compared to the substructures identified by baseline explainers, the subgraphs generated by MMHNN and MMGNN exhibit enhanced connectivity, a property attributed to their alternating and iterative selection processes that is absent in baseline explainers. For specific molecular nodes (e.g., N and O), certain neighboring nodes can introduce spurious signals, potentially leading to misleading explanations. MMGNN and MMHNN are explicitly designed to mitigate such issues, ensuring that the extracted explanations remain robust and interference-free. In contrast, baseline methods may include these confounding nodes, further underscoring the reliability and interpretability of MMHNN.

Generalization test on solvents

To evaluate the model's generalizability, a solvent holdout test (Fig. 4A) and a scaffold holdout test (Fig. 4B) are conducted by excluding specific solvents and scaffolds from the training and validation sets. MMHNN outperforms other models and has superior results on a variety of solvents, such as benzene (0.170 kcal mol^{-1}), ethanol (0.23 kcal mol^{-1}), hexane (0.12 kcal mol^{-1}), and acetone (0.225 kcal mol^{-1}), as shown in Table 6. However, the water solvent has the higher error compared to others. This difference could be intricately connected to factors such as the molecule's molar mass and element type.

The distribution of solute molar masses in each solvent, illustrated in Fig. 5A, provides a possible explanation. Most ΔG_{solv} data for polyatomic molecules are specifically measured in water, primarily as hydration Gibbs free energies. Molecular molar masses greater than 350 are predominantly present in aqueous solvents, constituting the test dataset for MMHNN. As a result, the model exhibits poor predictive performance due to the limited learning instances for these molecules. Clearly, octanol and water exhibit higher prediction errors compared to other solvents. This phenomenon is primarily attributed to the uneven distribution of solute molecules in the dataset. In particular, we found that the proportion of molecules with a molecular weight greater than 200 is higher in both octanol and water. This trend becomes even more pronounced for molecules exceeding 350 Da, compared to solvents such as ethanol (Fig. A1). This is reflected in the considerably higher test MAE for water solvent in Fig. 5A compared to other solvents.

Besides, we conducted an analysis of the relationship between the prediction errors of the model and the molar mass, and the number of atoms in the tested molecules in water solvent in Fig. 5B. Firstly, we could find that the model has excellent prediction results for molecules with a molar mass of less than 50, while

Table 5. Comparison of different explainability methods on various datasets. Underlined numbers are the best results and values are presented as mean (standard deviation).

Datasets	GIB-based	Global mask-based	Local mask-based	Gradient-based	Without explainability injection
Freesolv	0.527(0.028)	$0.536_{(0.010)}$	$0.547_{(0.028)}$	$0.602_{(0.010)}$	$0.651_{(0.010)}$
CompSol	0.144(0.011)	$0.146_{(0.010)}$	$0.154_{(0.015)}$	$0.163_{(0.017)}$	$0.175_{(0.024)}$
Abraham	0.183(0.007)	$0.187_{(0.010)}$	$0.189_{(0.020)}$	$0.205_{(0.026)}$	$0.328_{(0.047)}$
CompSol-Exp	$0.172_{(0.011)}$	$0.173_{(0.013)}$	$0.171_{(0.010)}$	0.146(0.010)	$0.208_{(0.015)}$
MNSol	0.269(0.013)	$0.281_{(0.011)}$	$0.305_{(0.012)}$	$0.326_{(0.024)}$	$0.342_{(0.032)}$

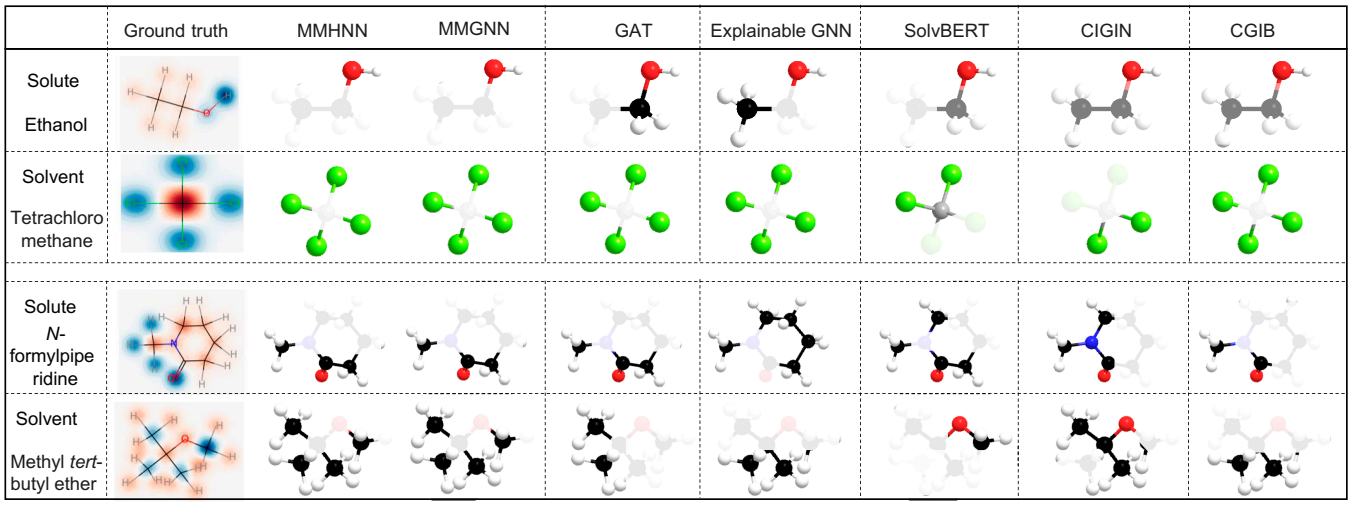


Fig. 3. Core substructures identified by different methods. Atoms with values above the average score are retained based on atomic weighted scores. In the ground truth, the color transition from pink to blue indicates an increasing significance of the corresponding atom.

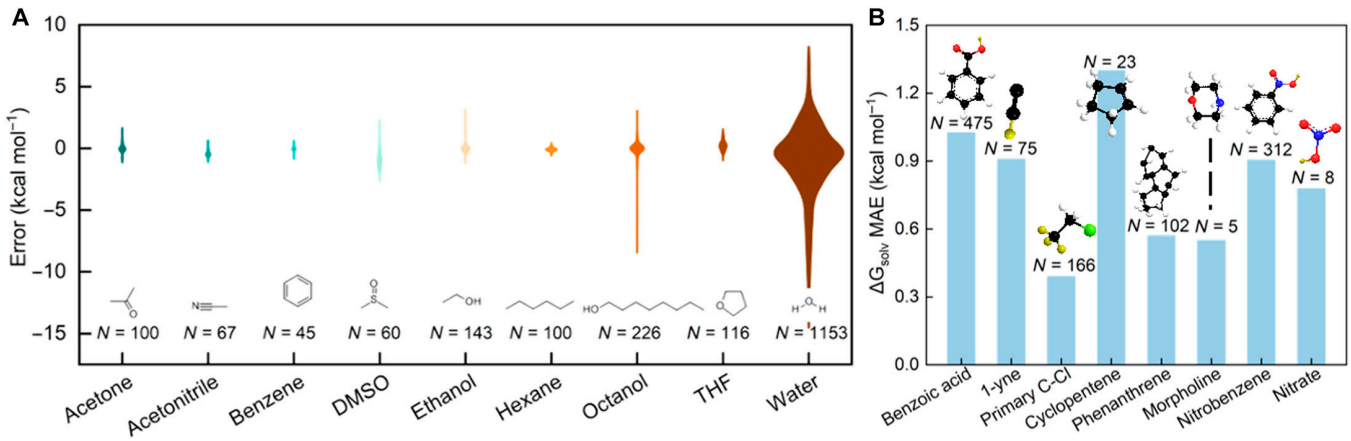


Fig. 4. (A) Violin plots of the MAE ($\Delta G_{pred} - \Delta G_{true}$) for various solvents including acetone, acetonitrile, benzene, and DMSO. (B) Results of the scaffold holdout test.

the prediction error of the MMHNN model exhibits a high correlation with the molecular molar mass, with an R^2 value of 0.65. This result suggests that the prediction error of the model may increase for larger molar mass molecules. However, a clear stratification phenomenon is observed in the predictive performance of MMHNN for molecules with molar masses between 50 and 350 in Fig. 5B, revealing a similar trend. This means that other factors may also play a significant role in affecting the model's performance in this molar mass range.

In addition, we analyzed the impact of elements, as the water solvent may contain more solutes with other elements besides C and H. We show the results of removing the test elements contained in different solute molecules in the water solvent system and the proportion of such elements in the total dataset in Fig. 5C. Clearly, the MMHNN model exhibits the poorest generalization performance for the element P, with an error of 5.55. However, it is worth noting that approximately 77.8% of P-containing molecules are found in aqueous solvent systems. Therefore, in the solvent holdout experiment, there are only 4 instances of P-containing molecules in both the training and validation sets. This limited representation in the data may be another contributing factor to the poor predictive performance of the model for aqueous solvents.

Generalization test on solute

To evaluate the model's generalization capabilities on a new category of solute molecules, we performed a scaffold-based split by assigning all solutes containing one or more functional groups or molecular scaffolds according to the MoleculeNet benchmark [28,36,37]. The prediction results for different molecular scaffolds vary significantly across various solvent systems (in Table 7). Among the 8 solvent systems tested, the water solvent system has the largest prediction error, which may be attributed to the complexity of the system and the presence of molecules containing a variety of elements such as phosphorus in the R group. The cyclopentene scaffolds had the worst predicted result in the octanol system, which was significantly worse than other molecular skeleton results. This difference might be attributed to the nonpolar nature of the cyclopentene scaffold as mentioned above. In summary, we meticulously examined and tested the generalization capabilities

Table 6. Cross-test results for various methods for solvents holdout test. Underlined numbers are the best results. Values are presented as mean (standard deviation).

	MAE (↓)					RMSE (↓)				
Solvent	GAT	Explainable GNN	CGIB	**MMGNN**	**MMHNN**	GAT	Explainable GNN	CGIB	**MMGNN**	**MMHNN**
Acetone	$0.301_{(0.032)}$	$0.284_{(0.023)}$	$0.252_{(0.024)}$	$0.231_{(0.012)}$	$0.225_{(0.010)}$	$0.372_{(0.022)}$	$0.355_{(0.021)}$	$0.342_{(0.024)}$	$0.331_{(0.022)}$	$0.327_{(0.020)}$
Acetonitrile	$0.592_{(0.051)}$	$0.482_{(0.023)}$	$0.454_{(0.023)}$	$0.425_{(0.030)}$	$0.421_{(0.027)}$	$0.532_{(0.051)}$	$0.484_{(0.041)}$	$0.477_{(0.040)}$	$0.462_{(0.041)}$	$0.460_{(0.038)}$
Benzene	$0.233_{(0.013)}$	$0.241_{(0.024)}$	$0.202_{(0.014)}$	$0.174_{(0.013)}$	$0.170_{(0.010)}$	$0.455_{(0.040)}$	$0.531_{(0.104)}$	$0.278_{(0.010)}$	$0.181_{(0.0217)}$	$0.179_{(0.015)}$
DMSO	$1.217_{(0.164)}$	$0.977_{(0.074)}$	$0.955_{(0.483)}$	$0.432_{(0.184)}$	$0.430_{(0.194)}$	$1.121_{(0.151)}$	$1.050_{(0.071)}$	$1.074_{(0.091)}$	$1.053_{(0.071)}$	$1.050_{(0.068)}$
Ethanol	$0.291_{(0.024)}$	$0.262_{(0.021)}$	$0.253_{(0.022)}$	$0.231_{(0.018)}$	$0.230_{(0.016)}$	$0.532_{(0.034)}$	$0.513_{(0.052)}$	$0.435_{(0.051)}$	$0.371_{(0.023)}$	$0.371_{(0.023)}$
Octanol	$0.452_{(0.023)}$	$0.461_{(0.045)}$	$0.374_{(0.028)}$	$0.321_{(0.024)}$	$0.319_{(0.023)}$	$0.842_{(0.034)}$	$0.854_{(0.042)}$	$0.827_{(0.031)}$	$0.825_{(0.031)}$	$0.823_{(0.030)}$
THF	$0.491_{(0.032)}$	$0.453_{(0.024)}$	$0.443_{(0.032)}$	$0.432_{(0.024)}$	$0.431_{(0.020)}$	$0.512_{(0.041)}$	$0.494_{(0.042)}$	$0.476_{(0.034)}$	$0.462_{(0.032)}$	$0.460_{(0.030)}$
Water	$2.584_{(0.045)}$	$2.324_{(0.037)}$	$2.117_{(0.043)}$	$2.105_{(0.052)}$	$2.100_{(0.049)}$	$3.804_{(0.119)}$	$3.204_{(0.120)}$	$3.197_{(0.104)}$	$3.162_{(0.107)}$	$3.160_{(0.100)}$
Hexane	$0.232_{(0.014)}$	$0.498_{(0.015)}$	$0.143_{(0.015)}$	$0.124_{(0.015)}$	$0.123_{(0.014)}$	$0.217_{(0.012)}$	$0.472_{(0.031)}$	$0.203_{(0.011)}$	$0.174_{(0.012)}$	$0.174_{(0.012)}$

of MMHNN, providing a solid foundation for the comprehensive assessment of our model's performance.

As shown in Fig. 4B, we considered 8 scaffolds including benzoic acid, 1-yne, primary C-Cl, cyclopentene, phenanthrene, morpholine, nitrobenzene, and nitrate as the test dataset. The remaining solutes were utilized for the training and validation datasets, ensuring that the solute compounds from the test set did not appear in the training or validation sets, and vice versa. This method of split is analogous to the commonly used Murcko scaffold splits in machine learning models for drug discovery. The scaffolds were chosen from a subgroup or subgraph as in previous studies [4].

In terms of predictive performance, heterocyclic and polycyclic molecules generally yield lower prediction errors compared to monocyclic counterparts, a trend that is also observed in the MMGNN model. For example, phenanthrene and morpholine—representative polycyclic and heterocyclic compounds—exhibit predictivity values of 0.57 and 0.56, respectively, whereas the monocyclic compound cyclopentene shows a significantly higher prediction error (1.31). We hypothesize that this discrepancy arises from the distinct physicochemical properties of these molecular classes: monocyclic molecules are more likely to be nonpolar, while polycyclic and heterocyclic compounds often possess polar functional groups. The presence of polar moieties, such as 1-yne and nitrate, appears to facilitate model learning, as these features contribute to more distinguishable spectral or structural signatures.

Overall, the MMHNN model exhibits strong generalization capability across diverse molecular scaffolds. Nonetheless, its predictive accuracy may be moderately reduced when extrapolating to unseen, nonpolar scaffolds that lack informative or distinctive features.

Complex system testing

Furthermore, to evaluate the model's performance on complex molecular systems, we selected the top 300 molecular pairs with the highest relative molecular weights from the dataset. These samples represent larger and more structurally intricate molecules. As shown in Fig. 6A, the selected pairs exhibit significantly higher molecular weights compared to the overall distribution in the 5 benchmark datasets (see Fig. A1). Elemental analysis of these complex molecules, shown in Fig. 6B, reveals a concentration around C, O, and F elements, with a notably lower proportion of nitrogen compared to the full dataset. This suggests that complex molecules in the current dataset tend to be carbon-rich organic compounds rather than biomolecules. Performance results are reported in Fig. 6C, where MMHNN achieves superior predictive accuracy. The model significantly outperforms baseline methods such as GAT, Explainable GNN, and CGIB. This demonstrates the advantage of the merged-graph strategy in modeling complex molecular systems, where atomic interactions are highly intricate and difficult to quantify. By assigning adaptive weights to relational edges, the fused graph better captures subtle interatomic dependencies. Compared to MMGNN, our model further reduces prediction error by approximately 7.16%. With the improved core substructure extraction mechanism, it more precisely identifies key atomic relationships, enabling accurate and robust predictions in challenging molecular scenarios.

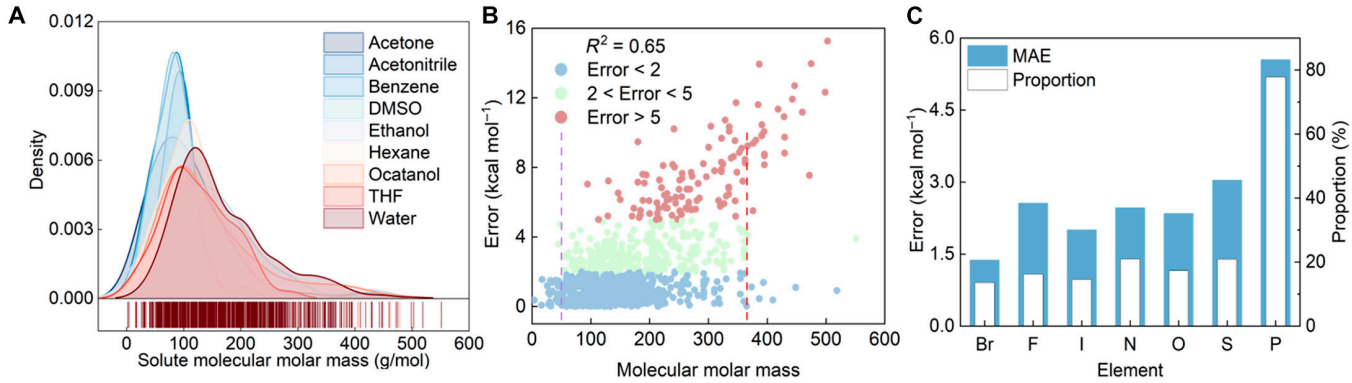


Fig. 5. (A) The test solute molecular molar mass distributions in various solvents. (B) The relationship between molar mass of different solutes and prediction error in water solvent. (C) The element holdout test in water solvents and proportion of test molecules containing such elements in datasets.

Table 7. Test results (MAE and RMSE results) for various scaffolds under different solvent conditions ("-" indicates that molecules with this scaffold do not exist in the solvent). Values are presented as mean (standard deviation).

MAE (↓)									
Solvent/Solute	Acetone	Acetonitrile	Benzene	DMSO	Ethanol	Hexane	Octanol	THF	Water
Benzoic acid	$0.27_{(0.02)}$	-	-	-	$0.27_{(0.01)}$	-	$0.13_{(0.01)}$	$0.25_{(0.01)}$	$2.32_{(0.05)}$
1-yne	$0.12_{(0.01)}$	-	-	$0.73_{(0.12)}$	-	$0.21_{(0.01)}$	-	-	$2.29_{(0.05)}$
Primary C-Cl	$0.21_{(0.01)}$	-	$0.13_{(0.01)}$	-	$0.40_{(0.03)}$	$0.16_{(0.01)}$	$0.10_{(0.01)}$	$0.28_{(0.02)}$	$0.68_{(0.03)}$
Cyclopentene	-	-	-	$0.67_{(0.10)}$	-	-	$0.39_{(0.08)}$	-	$1.77_{(0.04)}$
Phenanthrene	$0.13_{(0.01)}$	-	-	-	$0.07_{(0.01)}$	$0.10_{(0.01)}$	$0.12_{(0.01)}$	$0.41_{(0.03)}$	$2.16_{(0.07)}$
Morpholine	-	-	-	-	-	-	-	-	$2.32_{(0.08)}$
Nitrobenzene	$0.32_{(0.03)}$	$0.25_{(0.02)}$	-	-	$0.12_{(0.01)}$	$0.12_{(0.01)}$	$0.17_{(0.01)}$	$0.29_{(0.02)}$	$3.63_{(0.12)}$
Nitrate	-	-	-	-	-	-	-	-	$1.14_{(0.03)}$
Total	$0.23_{(0.02)}$	$0.42_{(0.03)}$	$0.17_{(0.01)}$	$0.95_{(0.18)}$	$0.23_{(0.02)}$	$0.12_{(0.01)}$	$0.32_{(0.02)}$	$0.43_{(0.03)}$	$2.10_{(0.05)}$
RMSE (↓)									
	Acetone	Acetonitrile	Benzene	DMSO	Ethanol	Hexane	Octanol	THF	Water
Benzoic acid	$0.34_{(0.02)}$	-	-	-	$0.69_{(0.05)}$	-	$0.21_{(0.01)}$	$0.31_{(0.02)}$	$3.00_{(0.12)}$
1-yne	$0.14_{(0.01)}$	-	-	$0.76_{(0.06)}$	-	$0.21_{(0.01)}$	-	-	$2.52_{(0.09)}$
Primary C-Cl	$0.22_{(0.01)}$	-	$0.13_{(0.01)}$	-	$0.40_{(0.02)}$	$0.20_{(0.01)}$	$0.13_{(0.01)}$	$0.37_{(0.03)}$	$0.89_{(0.07)}$
Cyclopentene	-	-	-	$0.67_{(0.06)}$	-	-	$4.50_{(0.12)}$	-	$2.01_{(0.07)}$
Phenanthrene	$0.17_{(0.01)}$	-	-	-	$0.08_{(0.01)}$	$0.11_{(0.01)}$	$0.13_{(0.01)}$	$0.45_{(0.03)}$	$2.34_{(0.10)}$
Morpholine	-	-	-	-	-	-	-	-	$2.39_{(0.10)}$
Nitrobenzene	$0.32_{(0.03)}$	$0.25_{(0.02)}$	-	-	$0.15_{(0.01)}$	$0.15_{(0.01)}$	$0.27_{(0.02)}$	$0.32_{(0.03)}$	$4.74_{(0.13)}$
Nitrate	-	-	-	-	-	-	-	-	$1.24_{(0.05)}$
Total	$0.33_{(0.02)}$	$0.46_{(0.04)}$	$0.18_{(0.01)}$	$1.07_{(0.09)}$	$0.37_{(0.02)}$	$0.17_{(0.01)}$	$0.82_{(0.03)}$	$0.46_{(0.03)}$	$3.16_{(0.10)}$

Complexity analysis

Constructing relationship edges in pairs, as previously done, has proven to be an effective strategy for predicting properties between molecular pairs. However, this approach significantly increases graph complexity, particularly for large, intricate molecules, due to the substantial rise in degrees of freedom. This leads to higher computational resource demands and increased time consumption. Motivated by the superior performance of the merged graph in molecular relationship learning, we are actively exploring advanced methods and techniques to reduce graph complexity. Currently, leveraging subgraph sets as hypernodes to construct a hypergraph is a promising approach. This method effectively reduces the complexity of the merged graph while maintaining the accuracy of molecular relationship learning as shown in Fig. 7. This design significantly reduces computational cost: model parameters decrease by approximately 90%, training time is reduced by 34%, and memory consumption drops by 6.25%, while predictive performance improves, with RMSE reduced by 3.62%. These results demonstrate that our approach not only simplifies the merged graph structure but also preserves—if not enhances—the accuracy of molecular relationship prediction.

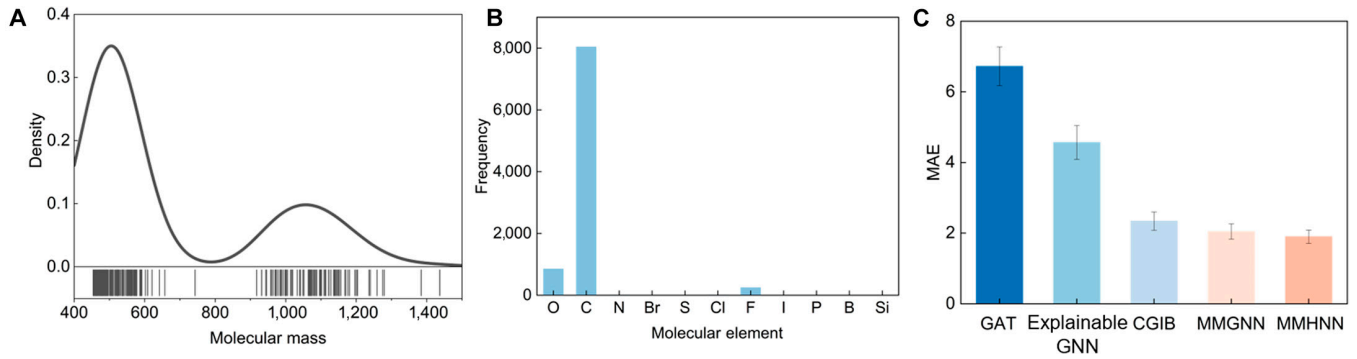


Fig. 6. (A) Molecular weight distribution of the selected complex molecular pairs. (B) Elemental composition of the selected molecules. (C) Model performance on complex molecular systems.

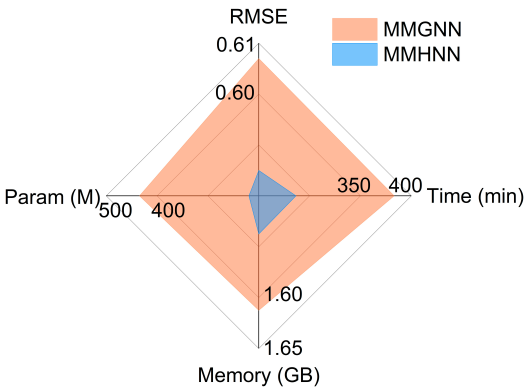


Fig. 7. Comparison of resource consumption between MMGNN and MMHNN.

Sensitivity analysis

We analyze the influence of the hyperparameter β (ranging from 0 to 0.1) on subgraph extraction, which controls the trade-off between prediction accuracy and information compression, as defined in Eq. 25. When $\beta = 0$, the model retains the full input graph without enforcing compression, leading to suboptimal performance due to the lack of structural abstraction. As β increases, performance initially improves as the model better identifies core substructures. However, beyond a certain threshold, further increasing β causes performance degradation. In particular, setting $\beta = 0.1$ results in a significant drop, likely due to excessive compression that removes critical input features. These results are summarized in Table 8.

Conclusion

In this study, we introduce MMHNN, a molecular graph neural network designed to predict solvation free energy. MMHNN leverages the complete connectivity between atoms within solute and solvent molecules, facilitating effective information transfer and enhancing model interpretability. Our experimental results demonstrate that MMHNN achieves superior performance across diverse datasets compared to existing methods, underscoring the importance of balancing intrinsic molecular properties and intermolecular interactions. The model also exhibits strong generalization capabilities, outperforming baselines in tasks such as solvent holdout and solute scaffold split. Moreover, MMHNN further enhances model performance while significantly reducing the complexity of the merged graph. The transparent interpretability of our model aligns well with established chemical principles, offering insights that resonate with experimental chemists. Additionally, a deeper understanding of molecular interactions provides valuable perspectives for assessing drug compound stability and guiding the development of novel synthetic pathways and catalytic processes.

Methods

In this section, we provide detailed descriptions of the implementation of each module.

Molecular graph representation

The molecular structure can be represented as a topological graph, where atoms correspond to nodes and bonds to edges [38,39]. Formally, the graph is defined as:

$$\mathcal{G} = \{\mathcal{E}, \mathcal{V}, \mathcal{U}\} \tag{1}$$

where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively, and $\mathcal{U}$ represents the global molecular features. In this formulation, all bonds within a molecule are explicitly enumerated. The edge set $\mathcal{E}$ consists of triplets (e_k, p_k, q_k), where e_k represents the edge features, while p_k and q_k denote the 2 atoms connected by

Table 8. Sensitivity analysis for β (RMSE indicator). Values are presented as mean (standard deviation).

	FreeSolv dataset				
2–6 β	0	1E−5	1E−4	1E−3	0.1
Result	$0.882_{(0.020)}$	$0.883_{(0.021)}$	$0.879_{(0.017)}$	$0.885_{(0.024)}$	$1.164_{(0.037)}$

bond e_k. The total number of bonds in the molecule is denoted as N^e:

$$\mathcal{E} = \left(e_k, p_k, q_k\right)_{k=1}^{N^e}, \quad \mathcal{V} = \left\{v_i\right\}_{i=1}^{N^v}. \tag{2}$$

Similarly, the node set $\mathcal{V}$ enumerates all atoms in the molecule, where v_i represents the atomic feature vector for atom i (e.g., atomic species, valence, etc.), and N^v is the total number of atoms. With these definitions, the molecular graph is fully specified.

For solute–solvent interactions, each dataset entry consists of a pair of molecular graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, representing the solute and solvent molecules, respectively, along with their corresponding scalar label, such as the Gibbs free energy of solvation $\mathbf{Y}$, where $\mathbf{Y} \in (-\infty, \infty)$.

Hypergraph construction

In order to complete the construction of the hypergraph and further realize the widely intermolecular message passing, we firstly need to complete the extraction of the substructure. Correspondingly, the subgraph set $\left\{\mathcal{G}_{sub1}^1, \ldots, \mathcal{G}_{sub1}^{c_1}\right\}$ and $\left\{\mathcal{G}_{sub2}^1, \ldots, \mathcal{G}_{sub2}^{c_2}\right\}$ from $\mathcal{G}_1$ and $\mathcal{G}_2$ will be obtained through the principal subgraph algorithm [40], where c_1 *and* c_2 represents the total number of subgraphs in $\mathcal{G}1$ and $\mathcal{G}2$ respectively. The extraction process is initialized by constructing the vocabulary V with all unique atoms, which are considered as subgraphs with a single node. This initialization can be represented as:

$$V = \left\{ v | v \in \bigcup_{\mathcal{G} \in \mathcal{D}} \mathcal{G} \right\}, \tag{3}$$

where $\mathcal{D}$ denotes the dataset containing all graphs.

During the merging phase, each pair of neighboring fragments F and F' in the current vocabulary is concatenated to create a new fragment. For each fragment pair $\left(F, F'\right)$, the merging operation can be described as:

$$F \cup F' \rightarrow F'', \tag{4}$$

where F'' is the new concatenated fragment. This operation is repeated iteratively.

The frequency of each concatenated subgraph is then counted, and the most frequent one is added to the vocabulary V. This updating process continues until the vocabulary size reaches a predefined number N. The frequency update for a subgraph S can be expressed as:

$$\text{frequency}\left(\mathcal{G}_{sub}\right) = \sum_{\mathcal{G}_i \in \mathcal{D}} \mathbb{I}\left(\mathcal{G}_{sub} \subseteq \mathcal{G}_i\right), \tag{5}$$

where $\mathbb{I}$ represents an indicator function that takes the value 1 if $\mathcal{G}_{sub}$ is a subgraph of $\mathcal{G}_i$ and 0 otherwise.

Through these steps, the subgraph sets $\left\{\mathcal{G}_{sub1}^1, \ldots, \mathcal{G}_{sub1}^{c_1}\right\}$ and $\left\{\mathcal{G}_{sub2}^1, \ldots, \mathcal{G}_{sub2}^{c_2}\right\}$ are effectively extracted from $\mathcal{G}_1$ and $\mathcal{G}_2$ using the principal subgraph algorithm. This subgraph extraction method is particularly representative for solute–solvent datasets because both solute and solvent molecules commonly share similar substructure types.

Once the subgraph sets are obtained, the original graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be transformed into hypergraphs, where each subgraph is treated as a hypernode. This transformation allows the complex structure of the original molecular graphs to be represented more compactly. The hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$ derived from $\mathcal{G}_1$ and $\mathcal{G}_2$ can be defined as:

$$\mathcal{H}_1 = \left\{\mathcal{G}_{sub1}^1, \mathcal{G}_{sub1}^2, \ldots, \mathcal{G}_{sub1}^{c_1}\right\}, \tag{6}$$

$$\mathcal{H}_2 = \left\{\mathcal{G}_{sub2}^1, \mathcal{G}_{sub2}^2, \ldots, \mathcal{G}_{sub2}^{c_2}\right\}. \tag{7}$$

In these hypergraphs, each hypernode represents a principal subgraph extracted from the original graph. Note that in these hypergraphs, edges do not carry meaningful information. During the intermolecular interaction process, we utilize hypernodes within the hypergraph for interaction.

Intramolecular message passing process

For the encoded solute–solvent molecular graph, we will perform L rounds of message passing, in which intramolecular message passing and intermolecular message passing mechanisms are performed alternately. We begin by performing message passing independently within each molecule—namely, the solute and solvent. During this intramolecular phase, each bond feature e_k is updated to a refined representation e'_{ij} by integrating the initial bond attributes, the features of the connected atoms v_i and v_j, and the global molecular descriptor u. Concurrently, the atomic representation v_i for each atom i is updated to a new vector v'_i.

$$e'_{ij} = e_{ij} + \tau\left[FC\left(v_i + v_j\right) + FC\left(e_{ij}\right) + FC(u)\right], \tag{8}$$

$$\hat{e}_{ij} = \frac{\sigma\left(e'_{ij}\right)}{\sum_{j' \in N_i} \sigma\left(e'_{ij'}\right) + \epsilon}, \tag{9}$$

$$v'_i = v_i + \tau\left[FC\left(v_i + \sum_{j \in N_i} \hat{e}_{ij} \odot FC\left(v_j\right)\right) + FC(u)\right], \tag{10}$$

where v'_i and e'_{ij} are the updated node vectors and edge vectors. τ is the LeakyReLU activation function [41]. FC is the fully connected layer. $\odot$ denotes the Hadamard product. ϵ is a fixed constant.

Intermolecular message passing process

Then, we describe the intermolecular messaging stage, which facilitates the exchange of information between hypernode $\left\{\mathcal{G}_{sub1}^1, \ldots, \mathcal{G}_{sub1}^{c_1}\right\}$ and $\left\{\mathcal{G}_{sub2}^1, \ldots, \mathcal{G}_{sub2}^{c_2}\right\}$ of hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$. This stage involves several key steps as follows.

First, for each hypernode $\mathcal{G}_{sub1}^i$ from hypergraph $\mathcal{H}_1$, we apply the Pooling operation to aggregate the features of its nodes $\mathcal{V}_{sub1}^i$. This operation is defined as:

$$\mathbf{H}_{sub1}^i = \text{Pooling}\left(\mathcal{V}_{sub1}^i\right), \tag{11}$$

where $\mathbf{H}_{sub1}^i$ represents the feature vector for hypernode $\mathcal{G}_{sub1}^i$.

Similarly, for each hypernode $\mathcal{G}_{sub2}^i$ from hypergraph $\mathcal{H}_2$, we perform the same Pooling operation:

$$\mathbf{H}_{sub2}^i = \text{Pooling}\left(\mathcal{V}_{sub2}^i\right), \tag{12}$$

where $\mathbf{H}^i_{sub2}$ represents the feature vector for hypernode $\mathcal{G}^i_{sub2}$.

Next, we compute the attention coefficients between each pair of hypernode feature vectors $\left(\mathbf{H}^i_{sub1},\mathbf{H}^j_{sub2}\right)$. These coefficients are calculated using the softmax function applied to the dot product of the feature vectors:

$$\alpha_{ij}=\frac{\exp\left(\mathbf{H}^i_{sub1}\cdot\mathbf{H}^j_{sub2}\right)}{\sum_k\exp\left(\mathbf{H}^i_{sub1}\cdot\mathbf{H}^k_{sub2}\right)}. \tag{13}$$

Using these attention coefficients, we update the feature vectors of the hypernodes by incorporating information from their corresponding pairs. The updated feature vectors are given by:

$$\mathbf{H}^i_{sub1}=\sum_j\alpha_{ij}\mathbf{H}^j_{sub2},\quad \mathbf{H}^j_{sub2}=\sum_i\alpha_{ij}\mathbf{H}^i_{sub1}. \tag{14}$$

Finally, we update the features of each node in the original graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ by incorporating the information from their corresponding hypernode feature vectors. For each node v in $\mathcal{V}_1^{(k)}$, the updated feature vector is computed as:

$$\mathcal{V}_1^{(k)}(v)=(1-\gamma)\mathcal{V}_1^{(k)}(v)+\gamma\mathbf{H}^i_{sub1},\text{where } v\in\mathcal{G}^i_{sub1}. \tag{15}$$

Similarly, for each node v in $\mathcal{V}_2^{(k)}$, the updated feature vector is given by:

$$\mathcal{V}_2^{(k)}(v)=(1-\gamma)\mathcal{V}_2^{(k)}(v)+\gamma\mathbf{H}^j_{sub2},\text{where } v\in\mathcal{G}^j_{sub2}. \tag{16}$$

where γ is a hyperparameter that controls the magnitude of message updates. This intermolecular message passing process effectively facilitates information transfer between hypergraphs, capturing the comprehensive features of solute–solvent molecular combinations, and ultimately enhancing the representation of the original molecular graphs $\mathcal{G}_1$ and $\mathcal{G}_2$.

Core hypernode extraction based on GIB

The GIB principle [42–44] is as follows:

$$\mathcal{G}_{\text{IB}}=\underset{\mathcal{G}_{\text{sub}}\in\mathcal{S}}{\arg\min}-I\left(\mathbf{Y};\mathcal{G}_{\text{sub}}\right)+\beta I\left(\mathcal{G};\mathcal{G}_{\text{sub}}\right). \tag{17}$$

Intuitively, $\mathcal{G}_{\text{IB}}$ is the core subgraph of $\mathcal{G}$, which discards information from $\mathcal{G}$ by minimizing the mutual information $I\left(\mathcal{G};\mathcal{G}_{\text{sub}}\right)$, while preserving target-relevant information by maximizing the mutual information $I\left(\mathbf{Y};\mathcal{G}_{\text{sub}}\right)$.

To detect the core reaction structure in hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$, we optimize the model with the objective function defined in Eq. 17 as follows:

$$\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}=\underset{\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}}{\arg\min}-I\left(\mathbf{Y};\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)+\beta\left(I\left(\mathcal{H}_1;\mathcal{H}_{\text{IB1}}\right)+I(\mathcal{H}_2;\mathcal{H}_{\text{IB2}})\right), \tag{18}$$

In the following sections, we provide the proof of upper bound of each term, which should be minimized during training.

Extract target-oriented information

To minimize $-I\left(\mathbf{Y};\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)$, we aim to calculate the upper bound of $-I\left(\mathbf{Y};\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)$. Given the hypergraph $\mathcal{H}_1$, $\mathcal{H}_2$, its label information $\mathbf{Y}$, and the learned IB-graph $\mathcal{H}_{IB1}$, $\mathcal{H}_{IB2}$, we have:

$$-I\left(\mathbf{Y};\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)\le\mathbb{E}_{\mathbf{Y};\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}}\left[-\log p_\theta\left(\mathbf{Y}|\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)\right]:=\mathcal{L}_{pre}, \tag{19}$$

where $p_\theta\left(\mathbf{Y}|\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)$ is the variational approximation of $p\left(\mathbf{Y}|\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)$. We model $p_\theta\left(\mathbf{Y}|\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)$ as a predictor parametrized by θ. Thus, we can minimize the upper bound of $-I\left(\mathbf{Y};\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)$ by minimizing the model prediction loss $\mathcal{L}_{\text{pre}}\left(\mathbf{Y},\mathcal{H}_{\text{IB1}},\mathcal{H}_{\text{IB2}}\right)$ with cross entropy loss. The proof is given in Appendix B.

Optimizing the minimization of $\mathcal{H}_1$

Minimizing $I\left(\mathcal{H}_1;\mathcal{H}_{\text{IB1}}\right)$, we aim to derive an upper bound of the mutual information term $I\left(\mathcal{H}_1;\mathcal{H}_{\text{IB1}}\right)$. We adopt a strategy by injecting noise into hypernode representations to suppress redundant information in $\mathcal{H}_1$ [43]. Specifically, we introduce noise sampled from a parameterized distribution and probabilistically replace hypernode features with this noise. For each hypernode i, a transmission probability p_i is learned via a fully connected (FC) layer applied to its embedding $\mathbf{H}^i_{sub1}$, followed by a sigmoid activation to ensure $p_i\in[0,1]$:

$$p_i=\text{Sigmoid}\left(FC\left(\mathbf{H}^i_{sub1}\right)\right). \tag{20}$$

Then, the hypernode representation $\mathbf{H}^i_{sub1}$ is stochastically replaced with noise ϵ based on the learned p_i. The resulting representation z_i is formulated as:

$$z_i=\lambda_i\mathbf{H}^i_{sub1}+\left(1-\lambda_i\right)\epsilon, \tag{21}$$

where $\lambda_i\sim\text{Bernoulli}\left(p_i\right)$ denotes a binary sampling variable that determines whether the original representation is preserved or replaced with noise. This stochastic gating mechanism enables fine-grained control over the information flow from $\mathbf{H}^i_{sub1}$ to z_i. When $p_i=1$, all information is retained; when $p_i=0$, it is fully replaced by noise. To enable gradient-based optimization despite the discrete nature of λ_i, we adopt the concrete relaxation [45], defined as:

$$\lambda_i=\text{Sigmoid}\left(\frac{1}{t}\log\frac{p_i}{1-p_i}+\log\frac{u}{1-u}\right), \tag{22}$$

where t is the temperature parameter and $u\sim\text{Uniform}(0,1)$. Another critical aspect of noise injection is the characterization of the injected noise. It is important that arbitrary noise can be detrimental to the semantic integrity of the input hypergraph, leading to predictions that deviate from the actual hypergraph properties. Conversely, appropriately selected noise can provide a variational upper bound to the overall objective. Therefore, the minimized upper bound of $I\left(\mathcal{H}_1;\mathcal{H}_{\text{IB1}}\right)$ is as follows:

$$I\left(\mathcal{H}_1;\mathcal{H}_{\text{IB1}}\right)\le\mathbb{E}_{\mathcal{H}_1}\left[-\frac{1}{2}\log A_{\mathcal{H}_1}+\frac{1}{2m_{\mathcal{H}_1}}A_{\mathcal{H}_1}+\frac{1}{2m_{\mathcal{H}_1}}B^2_{\mathcal{H}_1}\right]:=\mathcal{L}_{\text{MI}_1}\left(\mathcal{H}_{\text{IB1}},\mathcal{H}_1\right), \tag{23}$$

where $A_{\mathcal{H}_1}=\sum_{j=1}^{m_{\mathcal{H}_1}}\left(1-\lambda_j\right)^2$ and $B_{\mathcal{H}_1}=\frac{\sum_{j=1}^{m_{\mathcal{H}_1}}\lambda_j\left(\mathbf{H}^j_{sub1}-\mu_{\mathbf{H}}\right)}{\sigma_{\mathbf{H}}}$. A detailed proof is given in Appendix C.

Optimizing the minimization of $\mathcal{H}_2$

In order to minimize $I\left(\mathcal{H}_2;\mathcal{H}_{\mathrm{IB2}}\right)$, we aim to calculate the upper bound of $I\left(\mathcal{H}_2;\mathcal{H}_{\mathrm{IB2}}\right)$. Similarly, we have:

$$I\left(\mathcal{H}_2;\mathcal{H}_{\mathrm{IB2}}\right) \leq \mathbb{E}_{\mathcal{H}_2}[-\frac{1}{2}\log A_{\mathcal{H}_2}+ \frac{1}{2m_{\mathcal{H}_2}}A_{\mathcal{H}_2}+\frac{1}{2m_{\mathcal{H}_2}}B^2_{\mathcal{H}_2}] := \mathcal{L}_{\mathrm{MI}_2}\left(\mathcal{H}_2,\mathcal{H}_{\mathrm{IB2}}\right). \quad (24)$$

Loss function

In summary, our overall loss function is formulated by combining these individual components as follows, providing a comprehensive approach to optimizing our MMHNN for the intended molecular interaction tasks:

$$\mathcal{L}_{sum} = \mathcal{L}_{pre} + \beta\left(\mathcal{L}_{\mathrm{MI}_1} + \mathcal{L}_{\mathrm{MI}_2}\right), \quad (25)$$

where β is the trade-off parameter. $\mathcal{L}_{pre}$ is the binary cross entropy loss incorporating logits employed to mitigate prediction errors. $\mathcal{L}_{\mathrm{MI}_1}$ and $\mathcal{L}_{\mathrm{MI}_2}$ represent Kullback–Leibler divergence between extracted core subgraphs and random noise graphs, encouraging subgraph compression.

Acknowledgments

Funding: This paper is partially supported by the Project of Stable Support for Youth Team in Basic Research Field, CAS (YSBR-005), the Anhui Science Foundation for Distinguished Young Scholars (No. 1908085J24), the Natural Science Foundation of China (No. 62072427), and the Jiangsu Natural Science Foundation (No. BK2019119). The artificial intelligence (AI)-driven experiments, simulations, and model training were performed on the robotic AI-Scientist platform of the Chinese Academy of Sciences.

Author contributions: W.D. and S.Z. designed the work. W.D., Z.C., S.Z., and X.L. achieved the code. W.D., Z.C., S.Z., X.L., Z.L., J.F., J.W., and Y.W. prepared the original draft and drafted the original figures. All the authors read and approved the article.

Competing interests: The authors declare that they have no competing interests.

Data Availability

All of the data and code could be found in https://github.com/invokerqwer/MMHNN.

Supplementary Materials

The supplementary appendix provides the main theoretical proofs.

References

1. Varghese JJ, Mushrif SH. Origins of complex solvent effects on chemical reactivity and computational tools to investigate them: A review. *React Chem Eng*. 2019;4(2):165–206.
2. Fu Q, Ge J, Chen C, Wang Z, Yang F, Yin Y. High-precision colorimetric sensing by dynamic tracking of solvent diffusion in hollow-sphere photonic crystals. *Research*. 2022;2022:9813537.
3. D'Souza MJ, Darrington AM, Kevill DN. A study of solvent effects in the solvolysis of propargyl chloroformate. *ISRN Org Chem*. 2011;2011:767141.
4. Chung Y, Vermeire FH, Wu HY, Walker PJ, Abraham MH, Green WH. Group contribution and machine learning approaches to predict Abraham solute parameters, solvation free energy, and solvation enthalpy. *J Chem Inf Model*. 2022;62(3):433–446.
5. Fang J, Zhang S, Wu C, Yang Z, Liu Z, Li S, Wang K, Du W, Wang X, MolTC: Towards molecular relational modeling in language models. arXiv. 2024. https://doi.org/10.48550/arXiv.2402.03781
6. Low K, Coote ML, Izgorodina EI. Explainable solvation free energy prediction combining graph neural networks with chemical intuition. *J Chem Inf Model*. 2022;62(22):5457–5470.
7. Yu Y, Wang Z, Wang L, Wang Q, Tang R, Xiang S, Deng Q, Hou T, Sun H. Deciphering the shared and specific drug resistance mechanisms of anaplastic lymphoma kinase via binding free energy computation. *Research*. 2023;6:0170.
8. Zhang XC, Wu CK, Yi JC, Zeng XX, Yang CQ, Lu AP, Hou TJ, Cao DS. Pushing the boundaries of molecular property prediction for drug discovery with multitask learning BERT enhanced by SMILES enumeration. *Research*. 2022;2022:0004.
9. Liu Z, Qian W, Cai W, Song W, Wang W, Maharjan DT, Cheng W, Chen J, Wang H, Xu D, et al. Inferring the effects of protein variants on protein–protein interactions with interpretable transformer representations. *Research*. 2023;6:0219.
10. Wang X, Hsieh CY, Yin X, Wang J, Li Y, Deng Y, Jiang D, Wu Z, du H, Chen H, et al. Generic interpretable reaction condition predictions with open reaction condition datasets and unsupervised learning of reaction center. *Research*. 2023;6:0231.
11. Vermeire FH, Green WH. Transfer learning for solvation free energies: From quantum chemistry to experiments. *Chem Eng J*. 2021;418:Article 129307.
12. Lim H, Jung Y. MLSolvA: Solvation free energy prediction from pairwise atomistic interactions by machine learning. *J Cheminform*. 2021;13(1):56.
13. Hunter CA. Quantifying intermolecular interactions: Guidelines for the molecular recognition toolbox. *Angew Chem Int Ed Engl*. 2004;43(40):5310–5324.
14. Castner EW Jr, Wishart JF, Shirota H. Intermolecular dynamics, interactions, and solvation in ionic liquids. *Acc Chem Res*. 2007;40:1217–1227.
15. Yang L, Adam C, Nichol GS, Cockroft SL. How much do van der Waals dispersion forces contribute to molecular recognition in solution? *Nat Chem*. 2013;5(12):1006–1010.
16. Du W, Zhang S, Di Wu JX, Zhao Z, Fang J, Wang Y. MMGNN: A molecular merged graph neural network for explainable solvation free energy prediction. In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence 2024*. Jeju (South Korea); 2024. p. 5808–5816
17. Chi Z, Liu R, Yang B, Zhang H. Toxic interaction mechanism between oxytetracycline and bovine hemoglobin. *J Hazard Mater*. 2010;180(1–3):741–747.
18. Chen D, Gao K, Nguyen DD, Chen X, Jiang Y, Wei GW, Pan F. Algebraic graph-assisted bidirectional transformers for molecular property prediction. *Nat Commun*. 2021;12(1):3521.
19. Koh HY, Nguyen AT, Pan S, May LT, Webb GI. Physicochemical graph neural network for learning

protein–ligand interaction fingerprints from sequence data. *Nat Mach Intell*. 2024;6:673–687.
20. Bento AP, Hersey A, Felix E, Landrum G, Gaulton A, Atkinson F, Bellis LJ, De Vehi M, Leach AR. An open source chemical structure curation pipeline using RDKit. *J Chem*. 2020;12:51.
21. Mobley DL, Guthrie JP. FreeSolv: A database of experimental and calculated hydration free energies, with input files. *J Comput Aided Mol Des*. 2014;28(7):711–720.
22. Moine E, Privat R, Sirjean B, Jaubert JN. Estimation of solvation quantities from experimental thermodynamic data: Development of the comprehensive CompSol databank for pure and mixed solutes. *J Phys Chem Ref Data*. 2017;46(3):Article 033102.
23. Grubbs LM, Saifullah M, De La Rosa, Ye S, Achi SS, Acree WE Jr, Abraham MH. Mathematical correlations for describing solute transfer into functionalized alkane solvents containing hydroxyl, ether, ester or ketone solvents. *Fluid Phase Equilib*. 2010;298(1):48–53.
24. Marenich AV, Kelly CP, Thompson JD, Hawkins GD, Chambers CC, Giesen DJ, Winget P, Cramer CJ, Truhlar DG . Minnesota solvation database (MNSOL) version 2012. 2020.
25. Yu J, Zhang C, Cheng Y, Yang Y-F, She Y-B, Liu F, Su Q, Su A. SolvBERT for solvation free energy and solubility prediction: A demonstration of an NLP model for predicting the properties of molecular complexes. *Dig Discov*. 2023;2:409–421.
26. Meng F, Zhang H, Collins-Ramirez JS, Ayers PW. Something for nothing: Improved solvation free energy prediction with learning. 2023.
27. Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv. 2017. https://doi.org/10.48550/arXiv.1710.10903
28. Rong Y, Bian Y, Xu T, Xie W, Wei Y, Huang W, Huang J. Self-supervised graph transformer on large-scale molecular data. *Adv Neural Inf Process Syst*. 2020;33:12559–12571.
29. Zhou G, Gao Z, Ding Q, Zheng H, Xu H, Wei Z, Zhang L, Ke G. Uni-Mol: A universal 3D molecular representation learning framework. 2023.
30. Fang X, Liu L, Lei J, He D, Zhang S, Zhou J, Wang F, Wu H, Wang H. Geometry-enhanced molecular representation learning for property prediction. *Nat Mach Intell*. 2022;4:127–134.
31. Pathak Y, Laghuvarapu S, Mehta S, Priyakumar UD. Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York (USA); 2020. Vol 32, p. 873–880.
32. Lee N, Hyun D, Na GS, Kim S, Lee J, Park C. Conditional graph information bottleneck for molecular relational learning. arXiv. 2023. https://doi.org/10.48550/arXiv.2305.01520
33. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv. 2014. https://doi.org/10.48550/arXiv.1412.6980
34. Fang J, Wang X, Zhang A, Liu Z, He X, Chua TS. Cooperative explanations of graph neural networks. In: *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. Washington, DC (USA): ACM; 2023. p. 616–624.
35. Fang J, Liu W, Gao Y, Liu Z, Zhang A, Wang X, He X. Evaluating post-hoc explanations for graph neural networks via robustness analysis. In: *37th Conference on Neural Information Processing Systems*. New Orleans (LA); 2023.
36. Liu S, Wang H, Liu W, Lasenby J, Guo H, Tang J. Pre-training molecular graph representation with 3D geometry. In: *The 10th International Conference on Learning Representations*. Virtual; 2022.
37. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V. MoleculeNet: A benchmark for molecular machine learning. *Chem Sci*. 2018;9(2):513–530.
38. Wen M, Blau SM, Spotte-Smith EWC, Dwaraknath S, Persson KA. BonDNet: A graph neural network for the prediction of bond dissociation energies for charged molecules. *Chem Sci*. 2021;12(5):1858–1868.
39. Zhou Z, Huang Q, Lin G, Yang K, Bai L, Wang Y. Greto: Remedying dynamic graph topology-task discordance via target homophily. In: *The Eleventh International Conference on Learning Representations*. Kigali (Rwanda); 2023.
40. Kong X, Huang W, Tan Z, Liu Y. Molecule generation by principal subgraph mining and assembling. *Adv Neural Inf Process Syst*. 2022;35:2550–2563.
41. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. In: *Proceedings of the 30th International Conference on Machine Learning*. Atlanta (GA); 2013. Vol. 30, p. 3.
42. Yu J, Xu T, Rong Y, Bian Y, Huang J, He R. Graph information bottleneck for subgraph recognition. arXiv. 2020. https://doi.org/10.48550/arXiv.2010.05563
43. Yu J, Cao J, He R. Improving subgraph recognition with variational graph information bottleneck. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*. IEEE; 2022. p. 19374–19383.
44. Miao S, Liu M, Li P. Interpretable and generalizable graph learning via stochastic attention mechanism. In: *International Conference on Machine Learning*. PMLR; 2022. p. 15524–15543.
45. Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax. arXiv. 2016. https://doi.org/10.48550/arXiv.1611.01144