communications chemistry

A Nature Portfolio journal

Article

https://doi.org/10.1038/s42004-025-01683-z

MolGraph-xLSTM as a graph-based dual-level xLSTM framework for enhanced molecular representation and interpretability

Yan Sun[1,2], Yutong Lu[3], Yan Yi Li[3], Zihao Jing[2], Carson K. Leung[1] & Pingzhao Hu[1,2,3,4,5,6]

Predicting molecular properties is essential for drug discovery, and computational methods can greatly enhance this process. Molecular graphs have become a focus for representation learning, with Graph Neural Networks (GNNs) widely used. However, GNNs often struggle with capturing long-range dependencies. To address this, we propose MolGraph-xLSTM, a novel graph-based xLSTM model that enhances feature extraction and effectively models molecule long-range interactions. Our approach processes molecular graphs at two scales: atom-level and motif-level. For atom-level graphs, a GNN-based xLSTM framework with jumping knowledge extracts local features and aggregates multilayer information to capture both local and global patterns effectively. Motif-level graphs provide complementary structural information for a broader molecular view. Embeddings from both scales are refined via a multi-head mixture of experts (MHMoE), further enhancing expressiveness and performance. We validate MolGraph-xLSTM on 21 datasets from the MoleculeNet and Therapeutics Data Commons (TDC) benchmarks, covering both classification and regression tasks. On the MoleculeNet benchmark, our model achieves an average AUROC improvement of 3.18% for classification tasks and an RMSE reduction of 3.83% for regression tasks compared to baseline methods. On the TDC benchmark, MolGraph-xLSTM improves AUROC by 2.56%, while reducing RMSE by 3.71% on average. These results confirm the effectiveness of our model in learning generalizable molecular representations for drug discovery.

Predicting the molecular properties of a compound, particularly its ADMET (Absorption, Distribution, Metabolism, Excretion, and Toxicity) characteristics, is critical during the early stages of drug development[1,2]. Leveraging deep learning for molecular representation to predict these properties significantly enhances the efficiency of identifying potential drug candidates[3,4]. Molecular graphs retain richer structural information, which is crucial for accurate property prediction. In recent years, Graph Neural Networks (GNNs) built on molecular graph data have been extensively utilized for molecular representation learning to predict a wide range of properties[5–13].

A key challenge in molecular property prediction lies in capturing long-range dependencies—the influence of distant atoms or substructures within a molecule on a target property. While GNNs leverage neighborhood aggregation as their core mechanism—updating the hidden states of each node by aggregating information from neighboring nodes using operations like sum, max, or mean pooling[14,15]—they face significant limitations in capturing these long-range dependencies. Specifically, over-smoothing and over-squashing hinder their performance. Over-smoothing occurs when, as the number of layers increases, node representations become increasingly similar, leading to a loss of distinction between nodes[16]. On the other hand, over-squashing refers to the compression of information from distant nodes as it propagates toward the target node, making it challenging for relevant information to be effectively transmitted[17]. These issues limit the ability of GNNs to fully exploit global structural information, reducing their effectiveness in complex molecular property prediction tasks.

[1]Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada. [2]Department of Computer Science, Western University, London, ON, Canada. [3]Biostatistics Division, Dalla Lana School of Public Health, University of Toronto, Toronto, ON, Canada. [4]Department of Biochemistry, Western University, London, ON, Canada. [5]Department of Oncology, Western University, London, ON, Canada. [6]Department of Epidemiology and Biostatistics, Western University, London, ON, Canada. e-mail: phu49@uwo.ca

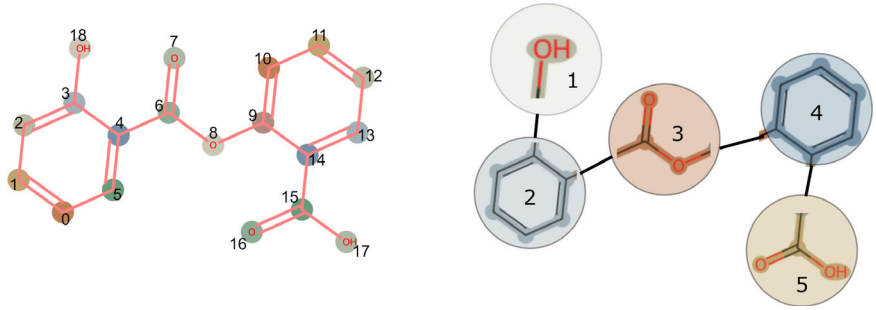

Fig. 1 | Comparison of atom-level and motif-level graph representations. a Atom-level graph representation, where each atom is represented as a node and each chemical bond as an edge. **b** Motif-level graph representation, where substructures are represented as single nodes, resulting in a graph that is less complex than the atom-level graph.

To address these challenges, we propose the MolGraph-xLSTM model, which integrates the extended Long Short-Term Memory (xLSTM) architecture with molecular graphs. Traditionally, Long Short-Term Memory (LSTM) networks have been widely used in Natural Language Processing (NLP) tasks to capture sequential data representations[18]. With its gating mechanisms, LSTM effectively decides which information to retain or discard, enabling it to manage long-range dependencies. Thus, we incorporate LSTM into our model to address the limitations of GNNs in handling long-range information. Recently, an improved version, xLSTM, was introduced[19]. xLSTM includes two additional modules, scalar Long Short-Term Memory (sLSTM) and matrix Long Short-Term Memory (mLSTM), which expand the storage capacity of the original LSTM. Experimental results have shown favorable performance compared to two state-of-the-art architectures: Transformer[20] and State Space Models[21]. For this reason, we chose this xLSTM model in our framework.

We utilize both atom-level and motif-level molecular graphs in our approach (Fig. 1). In the atom-level graph, each node represents an atom, and each edge represents a bond within the molecule. The motif-level graph, on the other hand, is a partitioned version of the atom-level graph, where each node represents a substructure (such as an aromatic ring) within a molecule. This results in a significantly simplified representation compared to the atom-level graph. Such simplification aids the model in learning features linked to local structures, as similar local motifs, from a functional group perspective, tend to impart similar properties to molecules[22]. Furthermore, the simplified motif-level graph, by reducing complexity and eliminating cycle structures, becomes closer to sequential data. This structural simplification aligns well with the strengths of xLSTM, which is inherently designed to handle sequential information, making the motif-level graph more suitable for processing with xLSTM.

However, relying solely on the motif-level graph would not capture all molecular details effectively, and motif partitioning itself demands precise segmentation. Therefore, we incorporate both atom-level and motif-level graphs in our model. For the atom-level representation, we introduce a GNN-based xLSTM with jumping knowledge[23]. Here, the GNN collects local information from the atom-level graph, and jumping knowledge aggregates features from multiple GNN layers, producing enriched node representations as inputs to xLSTM. By combining features from both the atom- and motif-level graphs, we constructed a comprehensive molecular representation for accurate property prediction.

Additionally, we integrate the Multi-Head Mixture-of-Experts (MHMoE) module[24] to enhance the predictive performance of our model. The sparse mixture-of-experts (SMoE)[25] framework has been demonstrated as an effective method for scaling models while maintaining computational efficiency by dynamically assigning inputs to different expert networks. This allows the input features to be processed by multiple experts, enabling diverse perspectives and improving the quality of learned representations. Building upon SMoE, the MHMoE architecture introduces further advancements by enhancing the usage of experts and promoting a more fine-grained understanding of input features. By incorporating the MHMoE module, our model is able to generate more expressive feature representations, which enhances its predictive accuracy.

The contributions of our work are as follows:

- Adaptation of xLSTM to dual-level molecular graph representation: We design a unified architecture that applies the xLSTM to both atom-level and motif-level molecular graphs. At the atom level, xLSTM follows GNN layers to enhance local features with long-range context. At the motif level, the graph is simplified through functional substructure decomposition, resulting in a sequential-like topology that further aligns with xLSTM's modeling strengths. This dual-level application enables comprehensive capture of fine-grained and high-level structural dependencies, substantially boosting prediction performance across 21 molecular property benchmarks.
- Integration of MHMoE for enhanced prediction: We incorporated the MHMoE module into our framework, which dynamically assigns input features to different expert networks, enabling diverse feature processing and improving predictive accuracy. This architecture refines feature representations through fine-grained expert activation.
- Case study analysis for model interpretability: We conducted a case study to investigate the substructures assigned the highest weights by the network, demonstrating that the atom-level and motif-level information are complementary. By cross-referencing with known literature, we identified strong correlations between the highlighted substructures and specific molecular properties, underscoring the ability of the model to implicitly learn biologically relevant information.

Results

Performance evaluation on MoleculeNet

MolGraph-xLSTM demonstrates improved performance across both classification and regression datasets, highlighting its robustness in handling diverse molecular property prediction tasks. In the classification tasks (Tables 1 and S1), MolGraph-xLSTM achieves particularly strong results on the Sider datasets. For the Sider dataset, MolGraph-xLSTM achieves an area under the receiver operating characteristic curve (AUROC) of 0.697 ± 0.022, representing a 5.45% improvement over the best baseline, FP-GNN (0.661 ± 0.014).

For regression datasets (Table 2 and Table S2), MolGraph-xLSTM delivers competitive performance across multiple benchmarks. On the ESOL dataset, MolGraph-xLSTM achieves a Root Mean Squared Error (RMSE) of 0.527 ± 0.046, reflecting a 7.54% improvement over the best-performing baseline, HiGNN (0.570 ± 0.061). On the FreeSolv dataset, MolGraph-xLSTM achieves the lowest RMSE of 1.024 ± 0.076 and the highest Pearson Correlation Coefficient (PCC) of 0.960 ± 0.006, demonstrating its reliability in regression tasks.

Performance evaluation on TDC benchmarks

MolGraph-xLSTM exhibits consistent performance across both classification and regression tasks in the TDC benchmark, indicating its capacity to generalize across diverse pharmacological endpoints. In classification tasks (Tables 3 and S3), MolGraph-xLSTM achieves the highest average AUROC (0.866) and area under the precision-recall curve (AUPRC) (0.861) across nine classification datasets, slightly outperforming competitive baselines

Table 1 | Performance evaluation on classification datasets from MoleculeNet

	Sider		Tox21		Clintox		BBBP		BACE		HIV	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
FP-GNN	0.661 ± 0.014	0.679 ± 0.026	0.833 ± 0.004	0.459 ± 0.018	0.732 ± 0.068	0.622 ± 0.028	0.892 ± 0.019	0.953 ± 0.007	0.852 ± 0.035	0.740 ± 0.042	0.767 ± 0.039	0.328 ± 0.078
DeeperGCN	0.622 ± 0.031	0.660 ± 0.025	0.840 ± 0.010	0.434 ± 0.021	0.892 ± 0.048	0.741 ± 0.048	0.860 ± 0.014	0.937 ± 0.008	0.830 ± 0.033	0.719 ± 0.039	0.769 ± 0.041	0.300 ± 0.064
DMPNN	0.658 ± 0.032	0.680 ± 0.030	0.849 ± 0.006	0.481 ± 0.026	0.895 ± 0.010	0.727 ± 0.062	0.896 ± 0.014	0.956 ± 0.016	0.851 ± 0.028	0.742 ± 0.043	0.758 ± 0.029	0.278 ± 0.043
HiGNN	0.656 ± 0.024	0.669 ± 0.027	0.844 ± 0.006	0.462 ± 0.018	0.889 ± 0.026	0.735 ± 0.070	0.892 ± 0.014	0.943 ± 0.017	0.836 ± 0.029	0.740 ± 0.048	0.768 ± 0.038	0.310 ± 0.066
TransFoxMol	0.636 ± 0.022	0.686 ± 0.040	0.816 ± 0.011	0.367 ± 0.011	0.830 ± 0.047	0.624 ± 0.036	0.881 ± 0.015	0.947 ± 0.005	0.801 ± 0.054	0.693 ± 0.079	0.727 ± 0.037	0.232 ± 0.063
BiLSTM	0.590 ± 0.008	0.631 ± 0.023	0.807 ± 0.011	0.382 ± 0.008	**0.987 ± 0.009**	**0.947 ± 0.023**	0.943 ± 0.020	0.985 ± 0.003	0.826 ± 0.052	**0.807 ± 0.076**	0.769 ± 0.040	0.305 ± 0.040
AutoML	0.682 ± 0.017	0.699 ± 0.029	0.828 ± 0.004	0.468 ± 0.022	0.875 ± 0.058	0.645 ± 0.018	0.928 ± 0.013	0.969 ± 0.019	0.840 ± 0.013	0.753 ± 0.018	0.772 ± 0.033	0.351 ± 0.078
MolGraph-xLSTM (Ours)	**0.697 ± 0.022**	**0.713 ± 0.032**	**0.854 ± 0.003**	**0.487 ± 0.045**	0.904 ± 0.032	0.714 ± 0.026	**0.959 ± 0.006**	**0.987 ± 0.002**	**0.869 ± 0.016**	0.784 ± 0.029	**0.775 ± 0.027**	**0.355 ± 0.050**

Best results are shown in bold.

such as DMPNN (AUROC: 0.861, AUPRC: 0.853) and FPGNN (AUROC: 0.859, AUPRC: 0.856).

MolGraph-xLSTM achieves noticeable improvement on the Bioavailability dataset, which measures the fraction of an administered drug that reaches systemic circulation. It obtains an AUROC of 0.684 ± 0.118, compared to 0.666 ± 0.035 from the best-performing baseline (FPGNN), and maintains a competitive AUPRC of 0.872 ± 0.057.

In regression tasks (Tables 4 and S4), MolGraph-xLSTM achieves leading or comparable results. It obtains the lowest RMSE on both the Caco2 (0.358 ± 0.015) and PPBR (11.772 ± 0.200) datasets, reflecting 11.17% and 3.81% improvements over the next-best models. Additionally, it achieves the highest PCC of 0.861 ± 0.011 on Caco2 and 0.644 ± 0.019 on PPBR.

Interpretability analysis

To evaluate the interpretability of MolGraph-xLSTM, we visualized the motifs and atomic sites with the highest model-assigned weights from the motif-level and atom-level networks. By applying max-pooling to the output of the xLSTM layer, we identified the features with the greatest contributions, providing us insight into the substructures and atomic sites that are most closely related to the properties of a particular molecule.

In Fig. 2, all three molecules highlight the—SO_2NH—(sulfonamide) substructure, a chemical motif known to be strongly linked with adverse reactions such as Type IV hypersensitivity, blurred vision, and other side effects[26]. These adverse effects correspond to side effects labeled in the Sider dataset, including Eye Disorders, Immune System Disorders, and Skin and Subcutaneous Tissue Disorders, demonstrating an alignment between the highlighted substructure and known biological properties of sulfonamides. Additionally, molecules like Fig. 2e, f emphasize atomic sites beyond the sulfonamide motif. In Fig. 2f, the highlighted N atom resides within the hydrazine group ($-NH-N=$), which is known to exert toxic effects on multiple organ systems, including neurological, hematological, and pulmonary[27]. This suggests that the atom-level network captures additional fine-grained features that complement the broader motif-level representations, demonstrating the capacity of the model to integrate complementary information from both atom-level and motif-level networks.

We further conducted an analysis on the BBBP dataset (blood-brain barrier permeability), a crucial property in evaluating the ability of a drug to cross the blood-brain barrier and target Central Nervous System (CNS) disorders. Accurate prediction of this property is essential for developing CNS-targeted therapies. For each molecule in the dataset, the substructure with the highest weight assigned by MolGraph-xLSTM was identified. These substructures were further analyzed using a random forest model[28] to determine their relationship with BBBP labels.

Fig. 3 illustrates the importance scores of substructures as determined by the random forest model. Among these, the substructure $-CC(=O)O-$, containing a carboxylic group ($-C(=O)O-$), achieved the highest importance score. This finding is supported by previous studies[29,30], which have highlighted the role of the carboxylic group in influencing BBBP.

Ablation study

Effect of different designed modules. We conducted an ablation study to evaluate the contributions of different components in MolGraph-xLSTM, including the atom-level branch (MolGraph-xLSTM (Atom-Level)), motif-level branch (MolGraph-xLSTM (Motif-Level)), multi-head mixture-of-experts module (MolGraph-xLSTM(w/o MHMoE)), and the GNN component within the atom-level branch (MolGraph-xLSTM (w/o GNN)). The results, presented in Table S5 and Fig. S1, highlight the importance of these components in achieving superior performance.

The full MolGraph-xLSTM model consistently outperformed all ablation variants, highlighting the effectiveness of its integrated architecture. Notably, even with only the atom-level branch, MolGraph-xLSTM achieved competitive performance, outperforming other atom-level graph-based models like DMPNN and DeeperGCN, as well as TransFoxMol, a hybrid model integrating GNN and Transformer. These results validate the design

Table 2 | Performance evaluation on regression datasets from MoleculeNet

	ESOL		Lipo		Freesolv	
	RMSE	PCC	RMSE	PCC	RMSE	PCC
FP-GNN	0.658 ± 0.006	0.946 ± 0.006	0.610 ± 0.028	0.861 ± 0.012	1.106 ± 0.195	0.951 ± 0.023
DeeperGCN	0.615 ± 0.044	0.954 ± 0.008	0.645 ± 0.048	0.842 ± 0.026	1.261 ± 0.022	0.938 ± 0.007
DMPNN	0.575 ± 0.073	0.957 ± 0.015	0.553 ± 0.033	0.842 ± 0.026	1.211 ± 0.120	0.945 ± 0.007
HiGNN	0.570 ± 0.061	0.959 ± 0.013	0.563 ± 0.041	0.882 ± 0.018	1.068 ± 0.092	0.956 ± 0.007
TransFoxMol	0.930 ± 0.261	0.917 ± 0.047	0.652 ± 0.033	0.855 ± 0.011	1.225 ± 0.155	0.945 ± 0.007
BiLSTM	0.743 ± 0.038	0.931 ± 0.012	0.779 ± 0.031	0.765 ± 0.026	1.398 ± 0.070	0.923 ± 0.015
AutoML	0.843 ± 0.062	0.910 ± 0.023	0.792 ± 0.043	0.748 ± 0.031	1.235 ± 0.220	0.941 ± 0.024
MolGraph- xLSTM (Ours)	**0.527 ± 0.046**	**0.965 ± 0.010**	**0.550 ± 0.026**	**0.888 ± 0.011**	**1.024 ± 0.076**	**0.960 ± 0.006**

Best results are shown in bold.

of our hybrid GNN and xLSTM framework as an effective approach for molecular representation learning. For the motif-level branch, it also outperformed other baselines on the Sider dataset, including HiGNN, which also utilizes motif-level graphs, in the classification task. However, its performance on the regression dataset was suboptimal. This suggests that the motif-level initialization features utilized in our model may not sufficiently capture the granularity required for regression tasks, highlighting opportunities for further improvement.

The MHMoE module contributed to the model performance, particularly on the FreeSolv dataset. Removing the MHMoE module resulted in an RMSE increase from 1.024 to 1.158, closely aligning with the performance of the atom-level-only variant, indicating its role in improving regression performance. As shown in Figs. S2 and S3, the activation maps demonstrate that all experts actively contribute to the task, indicating effective load balancing. This balanced activation ensures no single expert is overwhelmed, allowing the network to fully leverage the diverse expertise of all experts.

Among the four components, the GNN had the least impact on the Sider dataset but showed a notable influence on FreeSolv. Overall, the ablation study demonstrates that the atom- and motif-level branches provide complementary insights into molecular representation learning, and their integration enhances the model performance. This highlights the effectiveness of the proposed approach for molecular modeling.

Impact of node input order for molecular graphs on performance. xLSTM is originally designed for sequence data, which inherently has a fixed order. However, graph data does not have this property, as it can start from any node (Fig. 4). In our initial tests, we used the default node order provided by RDKit. In this section, we evaluate the effect of using a randomized starting node during training. Specifically, we generate the node sequence by performing a Depth-First Search (DFS) starting from a randomly selected initial node in the graph for each training instance.

Fig. S4 compares the performance of MolGraph-xLSTM trained with the RDKit default node order and the DFS random order on Sider and Freesolv datasets. On the Sider dataset (Fig. S4a), the model trained with the RDKit default order slightly outperformed the DFS random order in both AUROC and AUPRC metrics. Similarly, on the FreeSolv dataset (Fig. S4b), the RMSE and PCC metrics indicate a marginal advantage for the RDKit default order. Despite these differences, the results show that MolGraph-xLSTM achieves competitive performance with both node orderings. This suggests that the model is robust to changes in the input node sequence.

One possible explanation for this robustness is that although the initial node varies, the DFS imposes a relatively consistent traversal pattern across graphs. As a result, the relative positions of most nodes, particularly those within local substructures, tend to be preserved regardless of the starting point. This consistency likely helps maintain the stability of input sequences and contributes to the model's training stability and reproducibility across runs.

Long-range information retention via gate-based analysis

To provide direct evidence that the proposed xLSTM architecture captures long-range dependencies in the molecular graph, we performed a gate-based memory retention analysis. This analysis is based on the decay matrix $D[t, s]$, which measures how much the hidden state at timestep s contributes to the representation at timestep t through the internal gating mechanism of the model.

Formally, let i_k and f_k denote the input and forget gate activations at timestep k, respectively. The element $D[t, s]$ is defined as:

$$D[t,s] = \begin{cases} i_s \cdot \prod_{k=s+1}^{t} f_k, & 0 \le s \le t, \\ 0, & s > t, \end{cases}$$

where i_s determines how much new information is introduced at timestep s, and $\prod_{k=s+1}^{t} f_k$ quantifies the proportion of that information retained by the forget gates from $s+1$ to t. This formulation can be interpreted as a measure of temporal attention or memory retention within the xLSTM.

As an illustrative case study, we analyzed the molecule `C1=C[C@@H]([C@@H]2[C@H]1[C@@]3(C(=C([C@]2(C3(Cl)Cl)Cl)Cl)Cl)Cl)Cl` from the FreeSolv dataset. We examined $D[22, s]$, representing the influence of all previous timesteps $s \le 22$ on the final atom. The resulting memory retention plot is shown in Fig. 5.

Interestingly, the retention profile does not decay monotonically with temporal distance. Instead, multiple distant timesteps (e.g., steps 0-15) exhibit substantial influence, in some cases exceeding that of more recent steps. This suggests that xLSTM selectively preserves information from non-adjacent atomic contexts, adapting its retention patterns to the molecular structure and contextual requirements.

These findings provide direct evidence that xLSTM overcomes the short-range dependency bias inherent in standard GNNs, enabling effective modeling of non-local interactions across distant motifs or atoms.

Hyperparameter analysis

Performance of MolGraph-xLSTM with varying numbers of experts and heads in the MHMoE. The heatmaps in Fig. S5 reveal the impact of the number of experts and heads in the MHMoE module on the model's performance for the Sider and FreeSolv datasets. For both datasets, configurations with two experts generally perform poorly, while increasing the number of experts to 4 or 6 yields better results. Beyond 6 experts, no significant improvements are observed, suggesting that additional experts may become redundant for these datasets, as they do not process substantially different information.

For the Sider dataset, measured by AUROC, an increase in the number of heads consistently enhances performance, indicating that more heads improve the model's ability to handle classification tasks. In contrast, for the FreeSolv dataset, measured by RMSE, increasing the number of heads beyond 8 leads to a noticeable decline in performance, particularly when the

Table 3 | Performance evaluation on classification datasets from TDC

	HIA		Pgp		Bioavailability		CYP2D6-I		CYP3A4-I		CYP2C9-I		hERG		AMES		DILI	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
FPGNN	0.958 ± 0.012	0.985 ± 0.006	**0.930 ± 0.007**	**0.936 ± 0.008**	0.666 ± 0.035	**0.875 ± 0.012**	0.863 ± 0.005	0.641 ± 0.020	0.869 ± 0.004	0.840 ± 0.007	0.867 ± 0.002	0.720 ± 0.005	0.846 ± 0.029	0.929 ± 0.021	0.836 ± 0.006	0.878 ± 0.009	0.899 ± 0.017	0.896 ± 0.025
DeeperGCN	0.965 ± 0.017	0.990 ± 0.005	0.884 ± 0.009	0.900 ± 0.011	0.579 ± 0.113	0.811 ± 0.070	0.850 ± 0.005	0.613 ± 0.009	0.883 ± 0.004	0.852 ± 0.005	0.871 ± 0.002	0.741 ± 0.010	0.734 ± 0.050	0.869 ± 0.037	0.818 ± 0.010	0.861 ± 0.010	0.864 ± 0.026	0.865 ± 0.018
DMPNN	0.976 ± 0.004	**0.993 ± 0.001**	0.889 ± 0.005	0.904 ± 0.008	0.617 ± 0.050	0.849 ± 0.022	0.872 ± 0.004	0.644 ± 0.010	0.887 ± 0.003	0.858 ± 0.004	0.878 ± 0.004	0.750 ± 0.006	0.741 ± 0.024	0.857 ± 0.025	**0.838 ± 0.012**	**0.875 ± 0.013**	0.883 ± 0.021	0.872 ± 0.032
HiGNN	0.974 ± 0.007	0.991 ± 0.003	0.882 ± 0.022	0.885 ± 0.022	0.620 ± 0.070	0.850 ± 0.042	0.867 ± 0.007	0.633 ± 0.013	0.886 ± 0.006	0.853 ± 0.009	0.878 ± 0.009	0.741 ± 0.020	0.807 ± 0.027	0.918 ± 0.010	0.822 ± 0.014	0.856 ± 0.018	0.892 ± 0.011	0.870 ± 0.028
TransFoxMol	0.951 ± 0.036	0.983 ± 0.016	0.875 ± 0.011	0.890 ± 0.013	0.619 ± 0.019	0.840 ± 0.023	0.859 ± 0.004	0.603 ± 0.014	0.854 ± 0.009	0.827 ± 0.011	0.867 ± 0.004	0.721 ± 0.013	0.799 ± 0.017	0.907 ± 0.010	0.816 ± 0.011	0.860 ± 0.009	0.896 ± 0.024	**0.896 ± 0.013**
BiLSTM	0.893 ± 0.021	0.957 ± 0.012	0.890 ± 0.033	0.881 ± 0.035	0.605 ± 0.059	0.824 ± 0.021	0.834 ± 0.005	0.592 ± 0.012	0.829 ± 0.008	0.779 ± 0.014	0.852 ± 0.004	0.698 ± 0.009	0.843 ± 0.022	0.933 ± 0.014	0.747 ± 0.015	0.790 ± 0.015	0.826 ± 0.030	0.829 ± 0.020
AutoML	0.880 ± 0.012	0.933 ± 0.006	0.821 ± 0.007	0.769 ± 0.011	0.509 ± 0.026	0.761 ± 0.010	0.762 ± 0.011	0.444 ± 0.015	0.773 ± 0.004	0.652 ± 0.007	0.776 ± 0.010	0.558 ± 0.002	0.682 ± 0.021	0.814 ± 0.010	0.756 ± 0.008	0.760 ± 0.007	0.766 ± 0.017	0.710 ± 0.019
MolGraph-xLSTM (Ours)	**0.977 ± 0.010**	0.992 ± 0.003	0.908 ± 0.011	0.919 ± 0.009	**0.684 ± 0.118**	0.872 ± 0.057	**0.872 ± 0.002**	**0.659 ± 0.007**	**0.890 ± 0.005**	**0.863 ± 0.007**	**0.881 ± 0.006**	**0.757 ± 0.006**	**0.846 ± 0.017**	**0.933 ± 0.006**	0.832 ± 0.011	0.870 ± 0.010	**0.904 ± 0.006**	0.888 ± 0.006

Best results are shown in bold.

number of heads reaches 16. This decline is likely due to overfitting, as FreeSolv is a relatively small dataset. These observations highlight the need to balance the number of experts and heads based on the task and dataset size, as excessive complexity can negatively affect performance.

Performance of MolGraph-xLSTM with varying number of jump layers. The results in Fig. S6 illustrate the impact of varying the number of jump layers on the performance of MolGraph-xLSTM across the Sider and FreeSolv datasets. On the Sider dataset, the AUROC shows relatively small fluctuations, with the maximum value of 0.697 observed at 4 jump layers and the minimum value of 0.673 at 8 jump layers, representing a difference of 3.4%. In contrast, for the FreeSolv dataset, the impact of jump layers is more pronounced. The RMSE increases significantly from its lowest value of 1.042 at 4 jump layers to its highest value of 1.326 at 8 jump layers, a difference of 27%. The decline in performance at higher numbers of jump layers suggests that the inherent oversmoothing problem in GNNs may lead to the integration of overly smoothed deep features, which can negatively impact the performance of tasks requiring precise regression predictions.

Discussion

In this study, we propose a molecular representation learning framework that leverages xLSTM for both atom-level and motif-level graphs, providing a novel approach to molecular property prediction. Additionally, we incorporate the MHMoE module into our framework, which dynamically assigns input features to diverse expert networks, enhancing predictive accuracy through fine-grained feature activation. The effectiveness of our model is demonstrated across multiple molecular property prediction datasets, as presented in the "Results" section. Additional results for other evaluation metrics are provided in the supplementary material.

Our framework integrates atom-level and motif-level representations, and the ablation study highlights the independent effectiveness of these two levels. Specifically, both the atom-level and motif-level networks achieve competitive results individually in classification tasks (section "Effect of different designed modules"). However, the motif-level network exhibits a noticeable decline in regression performance. This limitation may be due to the initialization features of the motif-level graph, which rely on basic substructure properties, such as the counts of specific atoms (e.g., carbon) or bond types (e.g., single bonds). While these features capture useful information for classification tasks, they may lack the precision required for accurate regression predictions.

Regarding motif decomposition, certain complex molecules, such as polycyclic compounds with fused ring systems, can introduce structural complexity and pose challenges for decomposition. Nevertheless, the adopted decomposition strategy, ReLMole, applies uniform rules across all molecules, ensuring consistent motif representations regardless of topological intricacy. This consistency helps preserve the model's generalization ability, even when handling multi-ring systems.

We also note some trade-offs between different evaluation metrics. For example, while MolGraph-xLSTM generally achieves strong ranking-based performance across classification datasets, discrete metrics such as F1 or accuracy may be lower on certain datasets, reflecting conservative probability predictions near classification thresholds. Similarly, in regression tasks, RMSE and MAE values may show subtle differences, indicating the model's ability to control large errors while maintaining a centralized prediction distribution. These observations suggest opportunities for further calibration or representation refinement.

In addition to quantitative results, our interpretability analysis (section "Interpretability analysis") highlights the strengths of the model. By analyzing the high-weight substructures identified by the model, we observed biologically meaningful correlations between the recognized substructures and specific molecular properties. This demonstrates that the model not only achieves competitive predictive performance but also provides valuable interpretability. Such interpretability is crucial for practical applications, as it

Table 4 | Performance evaluation on regression datasets from TDC

	Caco2		PPBR		LD50	
	RMSE	PCC	RMSE	PCC	RMSE	PCC
FP-GNN	0.408 ± 0.056	0.835 ± 0.027	12.238 ± 1.194	0.614 ± 0.068	0.911 ± 0.040	0.544 ± 0.046
DeeperGCN	0.624 ± 0.034	0.470 ± 0.112	14.634 ± 0.392	0.305 ± 0.040	0.951 ± 0.063	0.472 ± 0.109
DMPNN	0.487 ± 0.103	0.796 ± 0.023	12.497 ± 0.230	0.588 ± 0.031	0.859 ± 0.035	0.608 ± 0.033
HiGNN	0.457 ± 0.064	0.794 ± 0.043	13.247 ± 0.724	0.554 ± 0.056	0.941 ± 0.038	0.523 ± 0.034
TransFoxMol	0.596 ± 0.082	0.719 ± 0.071	13.638 ± 0.349	0.512 ± 0.027	0.922 ± 0.053	0.538 ± 0.066
BiLSTM	0.611 ± 0.051	0.528 ± 0.103	13.930 ± 0.284	0.416 ± 0.041	0.980 ± 0.029	0.446 ± 0.038
AutoML	0.403 ± 0.014	0.820 ± 0.009	13.565 ± 0.139	0.471 ± 0.016	**0.841 ± 0.011**	**0.622 ± 0.012**
MolGraph-xLSTM (ours)	**0.358 ± 0.015**	**0.861 ± 0.011**	**11.772 ± 0.200**	**0.644 ± 0.019**	0.871 ± 0.026	0.600 ± 0.026

Best results are shown in bold.

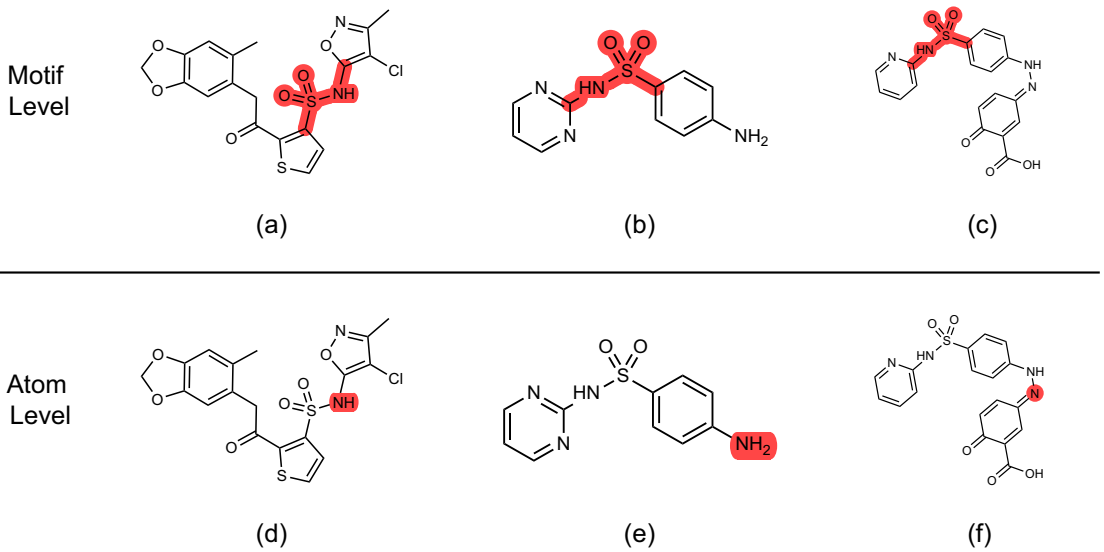

Fig. 2 | Visualization of the highest-weighted motifs and atoms identified by the model for molecules from the Sider test set containing the SO_2NH substructure. a–c Motifs with the highest attention weights from the motif-level branch. **d–f** Atoms with the highest attention weights from the atom-level branch.

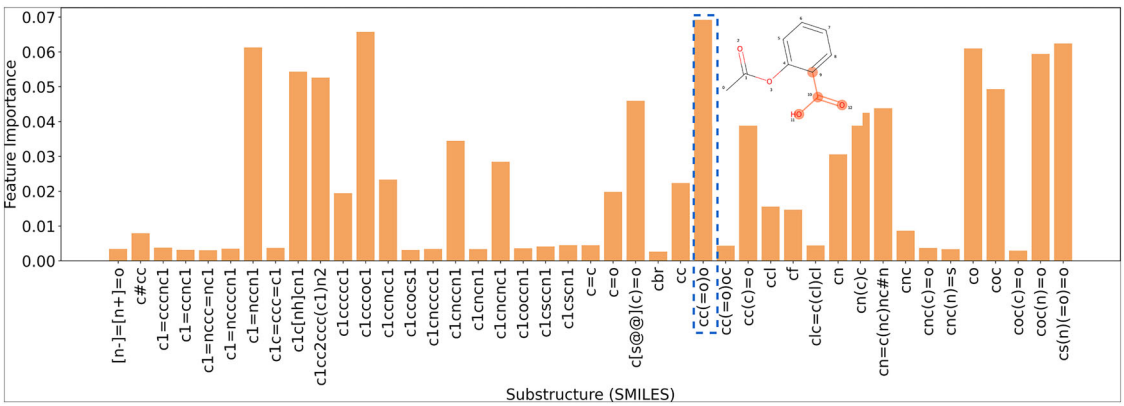

Fig. 3 | Importance scores of substructures identified by MolGraph-xLSTM on the BBBP dataset. For each molecule, the substructure with the highest model-assigned weight was analyzed using a random forest model to determine its relationship with BBBP labels. The substructure $-CC(=O)O-$, containing a carboxylic group, received the highest importance score (highlighted by the blue dashed box).

can assist in drug design by guiding the identification of key molecular features associated with desired properties.

To further assess the practical utility of our model, we provide a comparison of GPU memory usage, training time, and inference time with FP-GNN in the supplementary material (Table S6). These results indicate that, despite its architectural complexity, our model is computationally efficient in practice and well-suited for large-scale molecular screening tasks.

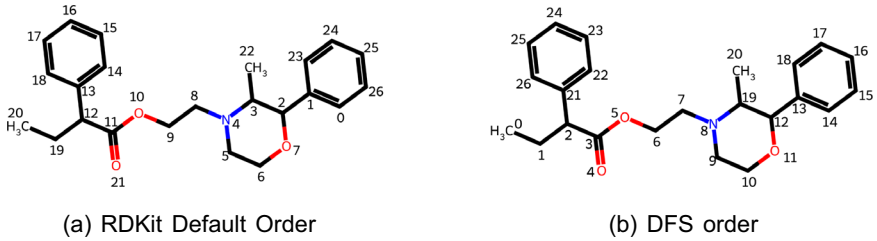

Fig. 4 | Examples of different atom input orders for molecular graphs in xLSTM. a RDKit default order: atoms are ordered as per the default output from RDKit. **b** DFS order: atoms are ordered based on a DFS traversal of the molecular graph.

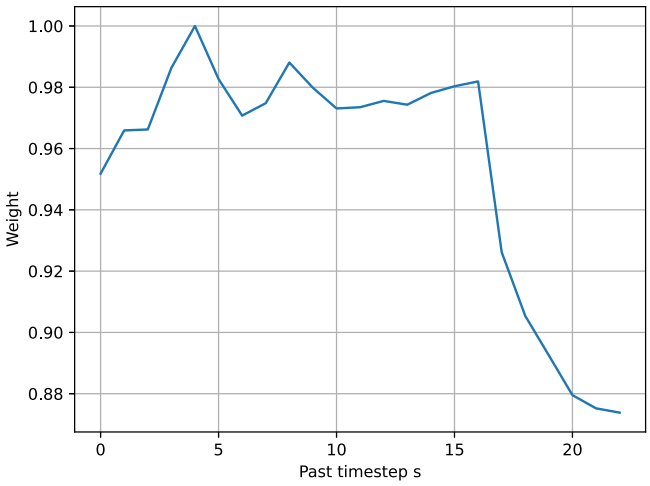

Fig. 5 | Memory retention. Gate-based memory retention plot at timestep 22 for the molecule `C1=C[C@@H](C[C@@H]2[C@H]1[C@@]3(C(=C(C[C@]2(C3(Cl)Cl)Cl)Cl)Cl)Cl)Cl` from the FreeSolv dataset.

Methods

Datasets and evaluation

MoleculeNet. MoleculeNet[31] is a widely used benchmark designed to evaluate machine learning models on molecular property prediction. We selected a subset of MoleculeNet datasets covering both classification and regression tasks.

For dataset splitting, we adopted different strategies based on task type. For single-task classification datasets, we employed scaffold splitting to ensure that molecules with different core scaffolds are separated into training, validation, and test sets. This strategy evaluates model generalization to novel chemical structures. For multi-task classification and regression datasets, we used random splitting to avoid data imbalance due to the relatively small dataset sizes.

Each dataset was split into training, validation, and test sets using an 8:1:1 ratio. The model was trained on the training set and evaluated on the validation set after each epoch. The best-performing model on the validation set was then used to report metrics on the test set. Each experiment was repeated three times, and we report the mean and standard deviation of the results.

Therapeutics data commons (TDC). We further evaluated our model on benchmark datasets from the TDC[32]. We adopted the official scaffold-based splits provided by TDC, where each dataset is partitioned into training, validation, and test sets in a 7:1:2 ratio. Each dataset includes five predefined splits. No additional resplitting or preprocessing was applied.

Evaluation metrics. For classification tasks, we used AUROC and AUPRC as evaluation metrics. For regression tasks, we reported RMSE and PCC. Detailed dataset information is summarized in Tables S7 and S8, and training hyperparameters are listed in Tables S9 and S10.

Hyperparameter tuning. For our proposed model, we performed grid search on the validation set to tune the hyperparameters, including the power coefficient (searched over {1, 2, 4}), hidden dimension ({64, 128, 256}), number of experts ({4, 8}), number of attention heads ({4, 8, 16}), and the number of expert layers ({1, 2, 3}). For baseline models, we followed the original implementations and used the reported hyperparameters when available; if not explicitly provided, we adopted values consistent with those used on similar datasets in the literature.

Baselines

We compare our proposed method against seven baseline models: Directed Message Passing Neural Network (DMPNN), Fingerprints and Graph Neural Networks (FPGNN), Hierarchical Informative Graph Neural Networks (HiGNN), Deeper Graph Convolutional Network (DeeperGCN), a transformer-based framework with focused attention (TransFoxMol), a sequence-based BiLSTM model, and an automated machine learning pipeline (AutoML). Each baseline represents a distinct approach to molecular representation learning or model optimization.

- FPGNN[10]: combines molecular fingerprints with features derived from graph attention networks, capturing both traditional cheminformatics features and structural insights from graphs.
- DeeperGCN[7]: a pure graph neural network based on GCN, designed for deeper architectures to enhance feature extraction.
- DMPNN[6]: optimizes message passing by centering aggregation on bonds instead of atoms, effectively encoding the chemical structure and avoiding redundant loops.
- HiGNN[33]: learns molecular representations at both the atomic level and the level of substructures using hierarchical GNNs.
- TransFoxMol[12]: integrates the power of GNNs and transformers to capture global and local molecular features efficiently.
- BiLSTM[34]: a sequence-based model that processes SMILES strings using Bidirectional LSTM layers to capture sequential molecular patterns.
- AutoML: a model selection and optimization pipeline based on automated machine learning techniques. It ensembles multiple algorithms and performs hyperparameter tuning automatically. In our experiments, we used H2O AutoML[35], which includes tree-based models such as XGBoost, Gradient Boosting Machine (GBM), and stacked ensembles.

xLSTM

A standard LSTM updates its cell state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$ through gated mechanisms:

$$\mathbf{c}_t = \mathbf{i}_t \odot \mathbf{z}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1}, \tag{1}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{2}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ denote the input, forget, and output gate vectors, respectively, and $\mathbf{z}_t$ is the candidate state vector. These gates are parameterized by *sigmoid* activations, which regulate information flow across time steps.

The xLSTM introduces two enhanced variants, sLSTM and mLSTM. Both replace the sigmoid gating functions in $\mathbf{i}_t$ and $\mathbf{f}_t$ with exponential gates, improving stability and extending effective memory:

$$\mathbf{i}_t = \exp(\mathbf{w}_i^\top \mathbf{x}_t + \mathbf{r}_i^\top \mathbf{h}_{t-1} + b_i), \tag{3}$$

$$\mathbf{f}_t = \exp(\mathbf{w}_f^\top \mathbf{x}_t + \mathbf{r}_f^\top \mathbf{h}_{t-1} + b_f), \tag{4}$$

where $\mathbf{w}_i$, $\mathbf{w}_f$, $\mathbf{r}_i$, and $\mathbf{r}_f$ are weight vectors, and b_i, b_f are bias scalars.

Furthermore, mLSTM extends the memory capacity by upgrading the vector-valued cell state $\mathbf{c}_t \in \mathbb{R}^d$ into a matrix-valued memory $\mathbf{C}_t \in \mathbb{R}^{d\times d}$, enabling richer storage and interactions:

$$\mathbf{C}_t = \mathbf{I}_t \odot \mathbf{Z}_t + \mathbf{F}_t \odot \mathbf{C}_{t-1}, \tag{5}$$

where $\mathbf{I}_t$, $\mathbf{F}_t$, and $\mathbf{Z}_t$ are matrix analogs of the input, forget, and candidate states, respectively.

The xLSTM block is formed by stacking alternating sLSTM and mLSTM layers, and multiple blocks are combined to construct the full xLSTM architecture. This design enhances the model's ability to capture long-range dependencies in sequential data.

Model architecture

Construction of atom- and motif-level molecular graphs. Starting from the SMILES string of a molecule, we convert it into an atom-level molecular graph $G_{\text{atom}} = \{V_{\text{atom}}, E_{\text{atom}}\}$ using the RDKit tool[36], where $V_{\text{atom}} = \{v_p^{\text{atom}}\}$ represents the set of nodes, and $E_{\text{atom}} = \{(v_p^{\text{atom}}, v_q^{\text{atom}})\}$ represents the set of edges. Each node v_p^{atom} corresponds to an atom and is initialized with 11 atomic features, including atomic number, chirality, and aromaticity (Table S11). Likewise, each edge $(v_p^{\text{atom}}, v_q^{\text{atom}})$ represents a bond and includes features such as bond type, stereochemistry, and conjugation (Table S12).

Based on the atom-level graph, we then generate a motif-level graph $G_{\text{motif}} = \{V_{\text{motif}}, E_{\text{motif}}\}$ through ReLMole, as described by ref. 37. In ReLMole, three types of substructures are considered as motifs: rings, non-cyclic functional groups, and carbon-carbon single bonds. In this motif graph, each node represents a motif and is initialized with 12 features, while each edge represents the connection between two motifs. Details of the initial features are provided in Table S13 in the Supplementary Information.

Both node and edge features are embedded into a d-dimensional feature vector. Specifically, we denote the input node feature matrix of the atom-level and motif-level graphs as $\mathbf{H}_{\text{atom}}^0 \in \mathbb{R}^{N_{\text{atom}}\times d}$ and $\mathbf{H}_{\text{motif}}^0 \in \mathbb{R}^{N_{\text{motif}}\times d}$, respectively, where N_{atom} is the number of atoms and N_{motif} is the number of motifs. The input feature vector of the edge in the atom-level graph between nodes p and q is $\mathbf{e}_{pq} \in \mathbb{R}^d$.

Feature extraction on the atom-level graph

Graph neural network. In the GNN component, we employ a simplified message-passing mechanism that incorporates both residual connections[7] and virtual nodes[14]. At each GNN layer, the process starts by applying Layer Normalization (LN) to the node representations, followed by a ReLU activation. To facilitate the exchange of global information across the graph, we introduce virtual nodes, which aggregate the features of all nodes in the graph. The resulting virtual node information is then added to the individual node representations. The operations can be formally expressed as:

$$\mathbf{h}_p^l = \text{ReLU}\left(\text{LN}(\mathbf{h}_p^l)\right) + \mathbf{v}^l, \tag{6}$$

$$\mathbf{v}^l = \sum_{k=1}^{N_{\text{atom}}} \mathbf{h}_k^l, \tag{7}$$

where $\mathbf{h}_p^l \in \mathbb{R}^d$ denotes the hidden state vector of node p at layer l, and $\mathbf{v}^l$ represents the virtual node vector.

Next, the message-passing step occurs, where the information from neighboring nodes and the edges connecting them is aggregated. For each edge $\mathbf{e}_{pq}$, a message is computed as:

$$\mathbf{m}_{pq} = \mathbf{h}_q^l + \mathbf{e}_{pq}.$$

The messages from all neighboring nodes $\mathcal{N}(p)$ are summed and used to update the node representation through an MLP:

$$\mathbf{h}_p^{l+1} = \text{MLP}\left(\sum_{q\in\mathcal{N}(p)} \mathbf{m}_{pq}\right). \tag{8}$$

Finally, a residual connection is applied, adding the original node representation from layer l to the updated node representation at layer $l+1$: $\mathbf{h}_p^{l+1} \leftarrow \mathbf{h}_p^{l+1} + \mathbf{h}_p^l$.

Jumping knowledge. After the GNN, we apply a jumping knowledge mechanism to aggregate information from all GNN layers. This allows each node feature to encapsulate representations from both shallow and deep layers. The operation is defined as:

$$\mathbf{h}_p^{\text{GNN}} = \text{CONCAT}\left(\mathbf{h}_p^1\mathbf{A}_1^T, \mathbf{h}_p^2\mathbf{A}_2^T, \ldots, \mathbf{h}_p^l\mathbf{A}_l^T\right), \tag{9}$$

where $\mathbf{h}_p^{\text{GNN}} \in \mathbb{R}^{d_{\text{skip}}\times n_{\text{jk}}}$ represents the aggregated feature vector of node p from the GNN, and $\mathbf{A}_l^T \in \mathbb{R}^{d\times d_{\text{skip}}}$ is a weight matrix that maps the layer-specific node feature $\mathbf{h}_p^l \in \mathbb{R}^d$ to a lower-dimensional space. In our experiments, we evaluate the impact of the number of jumping knowledge layers n_{jk} on performance.

Using xLSTM to capture long-range information. In this section, we utilize xLSTM to capture long-range dependencies for each node in the graph. We treat the output of the GNN, $\mathbf{H}^{\text{GNN}} \in \mathbb{R}^{N_{\text{atom}}\times(d_{\text{skip}}\times n_{\text{jk}})}$, as a sequence of length N_{atom}, where each row corresponds to one node. This sequence is then passed through the xLSTM model, producing an output $\mathbf{H}_{\text{atom}}^{\text{xLSTM}} \in \mathbb{R}^{N_{\text{atom}}\times(d_{\text{skip}}\times n_{\text{jk}})}$, as follows:

$$\mathbf{H}_{\text{atom}}^{\text{xLSTM}} = \text{xLSTM}(\mathbf{H}^{\text{GNN}}). \tag{10}$$

Motif-level feature extraction. The motif-level graph is processed directly by the xLSTM model. We first map the input feature $\mathbf{H}_{\text{motif}}^0$ to the dimension $d_{\text{skip}} \times n_{\text{jk}}$, matching the output dimension of the atom-level graph. This mapped feature is then passed through the xLSTM model to produce an output $\mathbf{H}_{\text{motif}}^{\text{xLSTM}} \in \mathbb{R}^{N_{\text{motif}}\times(d_{\text{skip}}\times n_{\text{jk}})}$:

$$\mathbf{H}_{\text{motif}}^{\text{xLSTM}} = \text{xLSTM}(\mathbf{H}_{\text{motif}}^0). \tag{11}$$

Perform a MHMoE on the features. We first apply a global max-pooling operation to $\mathbf{H}^{\text{GNN}}$, $\mathbf{H}_{\text{atom}}^{\text{xLSTM}}$, and $\mathbf{H}_{\text{motif}}^{\text{xLSTM}}$ to obtain three graph-level feature vectors: $\mathbf{f}_{\text{GNN}}$, $\mathbf{f}_{\text{atom}}^{\text{xLSTM}}$, and $\mathbf{f}_{\text{motif}}^{\text{xLSTM}}$. These are summed to produce the final molecular feature $\mathbf{f}_{\text{out}}$. Subsequently, an MHMoE module is applied to enhance representation learning.

For any input feature vector $\mathbf{f} \in \mathbb{R}^{h_{\text{moe}}\times d}$, we first partition it into h_{moe} segments $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{h_{\text{moe}}}$, each of dimension d. The output of the MoE layer for a given segment $\mathbf{f}_s$ is computed as:

$$\mathbf{f}_s^{\text{MoE}} = \sum_{i=1}^{n} G(\mathbf{f}_s)_i \, E_i(\mathbf{f}_s), \tag{12}$$

where E_i denotes the i-th expert, implemented as a feedforward network (FFN) with a configurable number of fully connected layers and nonlinear activations. The gating function $G(\mathbf{f}_s)_i$ assigns a weight to each expert:

$$G(\mathbf{f}_s) = \text{softmax}\left(\text{TopK}(g(\mathbf{f}_s) + \mathbf{D}_{\text{noise}}, K)\right), \tag{13}$$

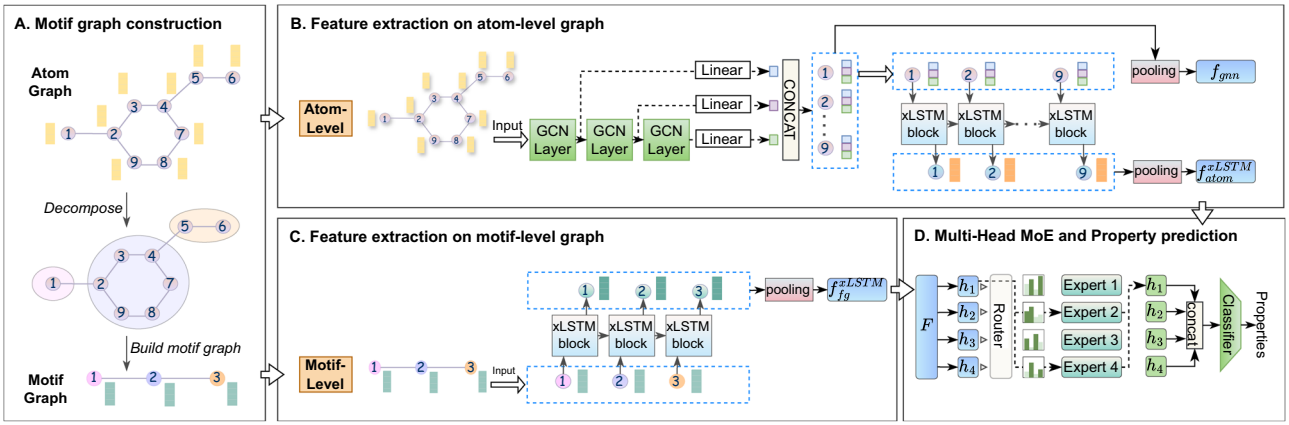

Fig. 6 | Architecture of MolGraph-xLSTM. The architecture consists of four main components: **A** motif graph construction. The atom-level graph is decomposed into motifs to form a motif-level graph. **B** Feature extraction on the atom-level graph. A GCN-based xLSTM framework with jumping knowledge extracts features, followed by pooling to generate the atom-level representation $\mathbf{f}_{\text{atom}}^{\text{xLSTM}}$. **C** Feature extraction on the motif-level graph. Using xLSTM blocks and pooling to produce the motif-level representation $\mathbf{f}_{\text{motif}}^{\text{xLSTM}}$. **D** MHMoE and property prediction. Features ($\mathbf{f}_{\text{gcn}}$, $\mathbf{f}_{\text{atom}}^{\text{xLSTM}}$ and $\mathbf{f}_{\text{motif}}^{\text{xLSTM}}$) are combined and refined through the MHMoE module for final property prediction.

where $g(\mathbf{f}_s)$ computes raw expert scores and $\mathbf{D}_{\text{noise}}$ introduces stochasticity during training. The TopK function selects the K highest-scoring experts:

$$\text{TopK}(\mathbf{v}, K)_i = \begin{cases} v_i & \text{if } v_i \text{ is among the top } K \text{ elements of } \mathbf{v}, \\ -\infty & \text{otherwise.} \end{cases} \tag{14}$$

Finally, the outputs of all segments are concatenated to form the MHMoE output:

$$\mathbf{f}^{\text{MHMoE}} = \text{CONCAT}(\mathbf{f}_1^{\text{MoE}}, \mathbf{f}_2^{\text{MoE}}, \ldots, \mathbf{f}_{h_{\text{moe}}}^{\text{MoE}}). \tag{15}$$

This design enables each segment of the input to be routed to the top-K most appropriate experts, allowing specialization of different experts for processing distinct types of molecular features.

Overall architecture. The overall architecture is illustrated in Fig. 6. We perform feature extraction on both the atom-level graph and the motif-level graph. For the atom-level graph, we first apply the GNN, followed by a skip connection that aggregates the outputs from all GNN layers, resulting in $\mathbf{H}^{\text{GNN}}$. This aggregated output is then passed through the xLSTM module, producing $\mathbf{H}_{\text{atom}}^{\text{xLSTM}}$ (section "Feature extraction on atom-level graph"). Next, global pooling is applied separately to $\mathbf{H}^{\text{GNN}}$ and $\mathbf{H}_{\text{atom}}^{\text{xLSTM}}$ to obtain graph-level features from the GNN ($\mathbf{f}_{\text{GNN}}$) and from the xLSTM ($\mathbf{f}_{\text{atom}}^{\text{xLSTM}}$). These two features are then summed to generate $\mathbf{f}_{\text{atom}} \in \mathbb{R}^{d_{\text{skip}} \times n_{\text{jk}}}$, representation of the feature of the atom-level graph.

The motif-level graph is fed directly into the xLSTM model, yielding $\mathbf{H}_{\text{motif}}^{\text{xLSTM}}$ (section "Motif-level feature extraction"). We obtain a graph-level feature $\mathbf{f}_{\text{motif}} \in \mathbb{R}^{d_{\text{skip}} \times n_{\text{jk}}}$ for the motif-level graph by applying global pooling on $\mathbf{H}_{\text{motif}}^{\text{xLSTM}}$. Then, $\mathbf{f}_{\text{atom}}$ and $\mathbf{f}_{\text{motif}}$ are summed to form the final molecular feature, which is passed through the MHMoE module (section "Perform a multi-head mixture-of-experts on the features") to further enhance the representation. Finally, the resulting feature is passed through an MLP to predict the molecular property:

$$\mathbf{f}_{\text{out}} = \text{MHMoE}(\mathbf{f}_{\text{atom}} + \mathbf{f}_{\text{motif}}), \tag{16}$$

$$\textbf{output} = \text{MLP}(\mathbf{f}_{\text{out}}), \tag{17}$$

where $\textbf{output} \in \mathbb{R}^K$, and K represents the number of tasks.

Loss function

To optimize the model, we applied two loss functions: the task loss $\mathcal{L}_{\text{task}}$ and the supervised contrastive loss (SCL) $\mathcal{L}_{\text{SCL}}$[38]. The task loss guides the model to minimize the error between the true label $\mathbf{y}$ and the predicted value $\hat{\mathbf{y}}$, while the SCL encourages the feature embeddings $\mathbf{f}_{\text{out}}$ to have samples with the same label close to each other in the embedding space, and samples with different labels far apart.

Task loss. For classification tasks, we use the cross-entropy loss, which measures the difference between the true label $\mathbf{y}_i$ and the predicted probability distribution $\hat{\mathbf{y}}_i$. This loss is formulated as:

$$\mathcal{L}_{\text{task}}^{\text{classification}} = -\sum_{k=1}^{K} y_{i,k} \log(\hat{y}_{i,k}), \tag{18}$$

where $y_{i,k}$ represents the true label for task k, and $\hat{y}_{i,k}$ is the predicted probability for task k.

For regression tasks, we adopt the Mean Squared Error (MSE) loss, which captures the discrepancy between the predicted value $\hat{y}_i$ and the true value y_i. The MSE loss is expressed as:

$$\mathcal{L}_{\text{task}}^{\text{regression}} = (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2. \tag{19}$$

SCL for the classification task. We apply the SCL to all features: $\mathbf{f}_{\text{out}}$, $\mathbf{f}_{\text{atom}}$, and $\mathbf{f}_{\text{motif}}$. Here, we illustrate the calculation using $\mathbf{f}_{\text{out}}$. First, we normalize $\mathbf{f}_{\text{out}}$ as:

$$\mathbf{f}_{\text{out}}^{\text{norm}} = \frac{\mathbf{f}_{\text{out}}}{\| \mathbf{f}_{\text{out}} \|_2 + \epsilon}, \tag{20}$$

$$\| \mathbf{f}_{\text{out}} \|_2 = \sqrt{\sum\nolimits_d f_{\text{out},d}^2}, \tag{21}$$

where ϵ is a small constant to prevent numerical instability, and d indexes the dimensions of the feature vector.

Next, the SCL $\mathcal{L}_{\text{SCL}}$ is computed using the normalized feature:

$$\mathcal{L}_{\text{SCL}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{f}_{\text{out},i}^{\text{norm}} \cdot \mathbf{f}_{\text{out},p}^{\text{norm}} / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{f}_{\text{out},i}^{\text{norm}} \cdot \mathbf{f}_{\text{out},a}^{\text{norm}} / \tau)}, \tag{22}$$

where i indexes the anchor molecule, $P(i)$ denotes the set of samples sharing the same label as the anchor, $A(i)$ represents the set of all sample indices excluding i, and τ is the temperature parameter.

SCL for the regression task. For regression tasks, positive samples are defined based on the Euclidean distance between labels of all sample pairs in the training set. Let d_{med} and $d_{\max}$ denote the median and maximum distances, respectively. A sample is considered positive for a given anchor if its distance to the anchor is less than d_{med}. Additionally, weights are assigned to reflect the relative importance of samples: closer positive samples are given higher weight, and farther negative samples are weighted more heavily. The SCL is formulated as:

$$\mathcal{L}_{\text{SCL}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} w_p \log \frac{\exp(\mathbf{f}_{\text{out},i}^{\text{norm}} \cdot \mathbf{f}_{\text{out},p}^{\text{norm}} / \tau)}{\sum_{a \in A(i)} w_a \exp(\mathbf{f}_{\text{out},i}^{\text{norm}} \cdot \mathbf{f}_{\text{out},a}^{\text{norm}} / \tau)}, \quad (23)$$

with weights defined as:

$$w_p = \frac{d_{\text{med}} - d_{ip}}{d_{\text{med}}}, \quad (24)$$

$$w_a = \exp\left(\frac{d_{ia} - d_{\text{med}}}{d_{\max} - d_{\text{med}}}\right), \quad (25)$$

where d_{ip} and d_{ia} are the Euclidean distances between sample i and sample p, and between sample i and sample a, respectively.

Overall loss function. The total loss for training the model is the sum of the task-specific loss and the SCL:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{SCL}}. \quad (26)$$

Data availability

The datasets used in this study are sourced from MoleculeNet (https://moleculenet.org/) and the TDC (https://tdcommons.ai/). The processed versions of these datasets used in our experiments are available on GitHub at https://github.com/syan1992/MolGraph-xLSTM/tree/main/datasets. The source data underlying all figures are provided in Supplementary Data 1.

Code availability

The source codes for MolGraph-xLSTM are freely available on GitHub at https://github.com/syan1992/MolGraph-xLSTM.

Received: 25 March 2025; Accepted: 29 August 2025;
Published online: 29 September 2025

References

1. Catacutan, D. B., Alexander, J., Arnold, A. & Stokes, J. M. Machine learning in preclinical drug discovery. *Nat. Chem. Biol.* **20**, 960–973 (2024).
2. Jia, L. & Gao, H. Machine learning for in silico admet prediction. *Artificial Intelligence in Drug Design* 447–460 (Methods in Molecular Biology, Clifton, 2022).
3. Jiménez-Luna, J., Grisoni, F. & Schneider, G. Drug discovery with explainable artificial intelligence. *Nat. Mach. Intell.* **2**, 573–584 (2020).
4. Sadybekov, A. V. & Katritch, V. Computational approaches streamlining drug discovery. *Nature* **616**, 673–685 (2023).
5. Xiong, Z. et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J. Med. Chem.* **63**, 8749–8760 (2019).
6. Yang, K. et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **59**, 3370–3388 (2019).
7. Li, G., Xiong, C., Thabet, A. & Ghanem, B. Deepergcn: all you need to train deeper gcns. Preprint at *arXiv* https://arxiv.org/abs/2006.07739 (2020).
8. Rong, Y. et al. Self-supervised graph transformer on large-scale molecular data. *Adv. Neural Inf. Process. Syst.* **33**, 12559–12571 (2020).
9. Wang, Y., Wang, J., Cao, Z. & Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
10. Cai, H., Zhang, H., Zhao, D., Wu, J. & Wang, L. Fp-gnn: a versatile deep learning architecture for enhanced molecular property prediction. *Brief. Bioinforma.* **23**, bbac408 (2022).
11. Fang, X. et al. Geometry-enhanced molecular representation learning for property prediction. *Nat. Mach. Intell.* **4**, 127–134 (2022).
12. Gao, J. et al. Transfoxmol: predicting molecular property with focused attention. *Brief. Bioinforma.* **24**, bbad306 (2023).
13. Zang, X., Zhao, X. & Tang, B. Hierarchical molecular graph self-supervised learning for property prediction. *Commun. Chem.* **6**, 34 (2023).
14. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Message passing neural networks. *Machine Learning Meets Quantum Physics* 199–214 (Springer, 2020).
15. Reiser, P. et al. Graph neural networks for materials science and chemistry. *Commun. Mater.* **3**, 93 (2022).
16. Li, Q., Han, Z. & Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proc. AAAI conference on artificial intelligence*, Vol. 32 (AAAI, 2018).
17. Alon, U. & Yahav, E. On the bottleneck of graph neural networks and its practical implications. In *Proc. 9th International Conference on Learning Representations* (ICLR, 2021).
18. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
19. Beck, M. et al. xlstm: extended long short-term memory. *Adv. Neural Inf. Process. Syst.* **37**, 107547–107603 (2024).
20. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems* 5998–6008 (NIPS, 2017).
21. Gu, A. & Dao, T. Mamba: linear-time sequence modeling with selective state spaces. Preprint at *arXiv* https://arxiv.org/abs/2312.00752 (2023).
22. Ertl, P. An algorithm to identify functional groups in organic molecules. *J. Cheminformatics* **9**, 1–7 (2017).
23. Xu, K. et al. Representation learning on graphs with jumping knowledge networks. In *Proc Machine Learning Research*, Vol. 8, 5449–5458 (ICML, 2018).
24. Wu, X. et al. Multi-head mixture-of-experts. *Adv. Neural Inf. Process. Syst.* **37**, 94073–94096 (2024).
25. Shazeer, N. et al. *Outrageously Large Neural Networks: the Sparsely-Gated Mixture-of-Experts Layer* (ICLR, 2017).
26. Daniel, D., Bacchi, S., Casson, R. & Chan, W. Sulfonamides in ophthalmology: adverse reactions: evidence-based use of sulfa drugs in ophthalmology. *Int. Ophthalmol.* **44**, 214 (2024).
27. Ivanov, I. & Lee, V. R. *Hydrazine Toxicology* (StatPearls Publishing, Treasure Island, 2023). http://europepmc.org/books/NBK592403.
28. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
29. Placzek, A. T. et al. Sobetirome prodrug esters with enhanced blood–brain barrier permeability. *Bioorg. Med. Chem.* **24**, 5842–5854 (2016).
30. Ferrara, S. J. & Scanlan, T. S. A cns-targeting prodrug strategy for nuclear receptor modulators. *J. Med. Chem.* **63**, 9742–9751 (2020).
31. Wu, Z. et al. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
32. Huang, K. et al. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Proc. Neural Information Processing Systems* (NeurIPS Datasets and Benchmarks, 2021).
33. Zhu, W., Zhang, Y., Zhao, D., Xu, J. & Wang, L. Hignn: a hierarchical informative graph neural network for molecular property prediction equipped with feature-wise attention. *J. Chem. Inf. Model.* **63**, 43–55 (2022).
34. Duy, H. A. & Srisongkram, T. Bidirectional long short-term memory (bilstm) neural networks with conjoint fingerprints: application in predicting skin-sensitizing agents in natural compounds. *J. Chem. Inf. Model.* **65**, 3035–3047 (2025).

35. LeDell, E. & Poirier, S. H2O AutoML: scalable automatic machine learning. In *Proc. AutoML Workshop at ICML* 24 (ICML, 2020).
36. Landrum, G. *Rdkit: Open-source Cheminformatics* (BibSonomy, 2006, accessed 7 January 2025). https://www.rdkit.org.
37. Ji, Z., Shi, R., Lu, J., Li, F. & Yang, Y. Relmole: molecular representation learning based on two-level graph similarities. *J. Chem. Inf. Model.* **62**, 5361–5372 (2022).
38. Khosla, P. et al. Supervised contrastive learning. *Adv. Neural Inf. Process. Syst.* **33**, 18661–18673 (2020).

Acknowledgements

This work was supported in part by the Canada Research Chairs Tier II Program (CRC-2021-00482), the Canadian Institutes of Health Research (PLL 185683, PJT 190272), the Natural Sciences and Engineering Research Council of Canada (RGPIN-2021-04072), and the Canada Foundation for Innovation (CFI) John R. Evans Leaders Fund (JELF) program (#43481).

Author contributions

Conceptualization: Y.S., Y.L., Y.Y.L., Z.J., and P.H. Investigation: Y.S., Y.L., Y.Y.L., Z.J., and P.H. Data curation: Y.S., Y.L., and Y.Y.L. Formal analysis: Y.S. Methodology development and design of methodology: Y.S. and P.H. Methodology creation of models: Y.S. Software: Y.S. Visualization: Y.S. Writing original draft: Y.S. Writing review editing: Y.S., Y.L., Y.Y.L., Z.J., P.H., and C.K.L. Funding acquisition: P.H. Supervision: P.H. and C.K.L.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42004-025-01683-z.

Correspondence and requests for materials should be addressed to Pingzhao Hu.

Peer review information *Communications Chemistry* thanks the anonymous reviewers for their contribution to the peer review of this work.