SCIENCE CHINA
Information Sciences

• Supplementary File •

Gelm: graph-based tanimoto similarity grouping pretraining and entropy-guided conformer selection finetuning for large language models

Zhuo Chen[1,2], Sihan Wang[1,2], Linjiang Chen[3], Wenjie Du[1,2*] & Yang Wang[1,2*]

[1]*School of Software Engineering, 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 215123, China*
[3]*State Key Laboratory of Precision and Intelligent Chemistry, USTC, Hefei, Anhui 230026, China*

Appendix A Datasets

In this section, we provide detailed information about the datasets used during training. We utilize a total of 12 datasets from a variety of domains, including DDI, SSI, and CSI. These datasets cover a wide range of molecular entities, offering a comprehensive foundation for our experiments. The specific details of the datasets are summarized in Table A1.

Task	Dataset	$\mathcal{G}^1$	$\mathcal{G}^2$	Pairs
DDI	DrugBank	1704	1704	191,400
	ZhangDDI	548	548	48,548
	ChChMiner	1322	1322	48,514
	DeepDDI	-	-	192,284
	TWOSIDES	555	555	3,576,513
SSI	MNSol	372	86	2,275
	FreeSolv	560	1	560
	CompSol	442	259	3,548
	Abraham	1038	122	6,091
	CombiSolv	1,368	291	10,145
	CombiSolv-QM	11,029	284	1,000,000
CSI	Chromophore	7,016	365	20,236

Table A1 Data statistics.

In the DDI task, $\mathcal{G}^1$ and $\mathcal{G}^2$ represent the number of drugs; in the SSI task, $\mathcal{G}^1$ and $\mathcal{G}^2$ represent the number of solutes and solvents, respectively; in the CSI task, $\mathcal{G}^1$ represents the number of chromophores and $\mathcal{G}^2$ represents the number of solvents. Pairs represent the total number of interaction pairs between the two categories. Our experiments involve a total of over 4,000,000 molecular pairs.

Appendix B Baselines

We employ both conventional deep learning models and SOTA biochemical LLMs as baselines. For qualitative tasks:

IGIB-ISE [1] integrates IGIB theory to refine molecular substructures for property prediction.
MHCADDI [2] uses a gated information transfer neural network and attention mechanism for substructure extraction and interaction modeling.
DeepDDI [3] computes structural similarity profiles and predicts interactions with a deep neural network.
SSI-DDI [4] extracts substructures using GAT network, with final predictions via co-attention.
CGIB [5] uses conditional information bottleneck theory to extract conditional subgraphs.
CMRL [6] identifies core substructures related to chemical reactions using a conditional intervention framework.
MDF-SA-DDI [7] uses multi-source drug and feature fusion with self-attention mechanism for predictions.
DSN-DDI [8] applies iterative learning for intra and inter-view substructure learning in drug interactions.

For quantitative tasks, we use the following baselines:
D-MPNN [9] combines quantum and experimental data for solvation free energy prediction.
SolvBert [10] predicts solvation free energy by fine-tuning models on solute-solvent interaction data.
SMD [11] uses quantum charge density and solvent continuum models to predict solvation free energy.
MMGNN [12]employs interpretable GNN to extract key subgraphs for solvation free energy prediction.
GEM [13] uses geometry-based GNN with self-supervised learning to predict molecular properties.
GROVER [14] extracts structural data from unlabeled molecular data using self-supervised tasks with a Transformer-style architecture.
Uni-Mol [15] features pre-trained SE(3) Transformer models for molecular and protein pocket data, with fine-tuning strategies for diverse tasks.

* Corresponding author (email: duwenjie@mail.ustc.edu.cn, angyan@ustc.edu.cn)

Appendix C Mathematical proof

In this section, we provide a complete derivation of the optimization objective in Equation 6 and its tractable form in Equation 12, based on the information bottleneck principle.

Appendix C.1 From graph information bottleneck to kc-subgraph objective

Given a conformation relation graph G^a and its selected subgraph G^a_{sub}, the goal is to retain task-relevant information while discarding redundant structural information. Following the information bottleneck principle, we aim to maximize the mutual information between the selected subgraph and the prediction target Y, while minimizing the dependence on the original graph:

$$\max_{G^a_{sub}} I(Y, G^{ab}_{sub}) - \alpha I(G^a, G^a_{sub}), \tag{C1}$$

where G^{ab}_{sub} denotes the fused representation of the selected subgraphs from the two molecules.

To further encourage structural compactness and suppress redundant conformers, we introduce a structural entropy regularizer $SE(G^a_{sub})$. The final objective becomes:

$$\max_{G^a_{sub} \in \mathcal{G}^a_{sub}} I(Y, G^{ab}_{sub}) - \alpha I(G^a, G^a_{sub}) + \beta SE(G^a_{sub}). \tag{C2}$$

This objective enforces three properties, predictive sufficiency, information compression and structural compactness.

Appendix C.2 Variational lower bound of $I(Y, G^{ab}_{sub})$

By definition, the mutual information is:

$$I(Y, G^{ab}_{sub}) = \mathbb{E}_{p(y, G^{ab}_{sub})} \left[\log \frac{p(y|G^{ab}_{sub})}{p(y)} \right]. \tag{C3}$$

This can be rewritten as:

$$I(Y, G^{ab}_{sub}) = \mathbb{E}_{p(y, G^{ab}_{sub})}[\log p(y|G^{ab}_{sub})] + H(Y), \tag{C4}$$

where $H(Y)$ is constant w.r.t. model parameters.

Introducing a variational distribution $q_{\phi_1}(y|G^{ab}_{sub})$, we have:

$$\log p(y|G^{ab}_{sub}) = \log q_{\phi_1}(y|G^{ab}_{sub}) + \log \frac{p(y|G^{ab}_{sub})}{q_{\phi_1}(y|G^{ab}_{sub})}. \tag{C5}$$

Taking expectation yields:

$$\begin{aligned} I(Y, G^{ab}_{sub}) = {} & \mathbb{E}_{p(y, G^{ab}_{sub})}[\log q_{\phi_1}(y|G^{ab}_{sub})] \\ & + \mathbb{E}_{p(G^{ab}_{sub})} \left[\mathrm{KL}(p(y|G^{ab}_{sub}) \| q_{\phi_1}(y|G^{ab}_{sub})) \right] + H(Y). \end{aligned} \tag{C6}$$

Since KL divergence is non-negative:

$$I(Y, G^{ab}_{sub}) \geqslant \mathbb{E}_{p(y, G^{ab}_{sub})}[\log q_{\phi_1}(y|G^{ab}_{sub})]. \tag{C7}$$

Using empirical samples:

$$\mathbb{E}[\cdot] \approx \frac{1}{N} \sum_{i=1}^{N} \log q_{\phi_1}(y_i|G^{ab}_{sub,i}). \tag{C8}$$

Thus, maximizing $I(Y, G^{ab}_{sub})$ is equivalent to minimizing:

$$L_{cls} = -\frac{1}{N} \sum_{i=1}^{N} \log q_{\phi_1}(y_i|G^{ab}_{sub,i}), \tag{C9}$$

which corresponds to cross-entropy loss for classification tasks.

Appendix C.3 Donsker-varadhan representation of $I(G^a, G^a_{sub})$

The second term is:

$$I(G^a, G^a_{sub}) = \mathrm{KL}\big(p(G^a, G^a_{sub}) \,\|\, p(G^a) p(G^a_{sub})\big). \tag{C10}$$

Direct computation is intractable. We adopt the Donsker–Varadhan (DV) representation:

$$I(G^a, G^a_{sub}) = \sup_{f_{\phi^a_2}} \Big(\mathbb{E}_{p(G^a, G^a_{sub})}[f_{\phi^a_2}] - \log \mathbb{E}_{p(G^a) p(G^a_{sub})}[e^{f_{\phi^a_2}}] \Big), \tag{C11}$$

where $f_{\phi^a_2}$ is a learnable function.

In practice, $f_{\phi_2^a}$ is parameterized by a neural network:

$$f_{\phi_2^a}(G^a, G^a_{sub}) = \mathrm{MLP}(\mathrm{GIN}(G^a) \,\|\, \mathrm{GIN}(G^a_{sub})). \tag{C12}$$

The empirical estimator becomes:

$$L_{MI}(\phi_2^a, G^a_{sub}) = \frac{1}{N}\sum_{i=1}^{N} f_{\phi_2^a}(G^a_i, G^a_{sub,i}) - \log\left(\frac{1}{N}\sum_{i=1}^{N}\sum_{j\neq i} e^{f_{\phi_2^a}(G^a_i, G^a_{sub,j})}\right). \tag{C13}$$

This provides a lower bound of $I(G^a, G^a_{sub})$.

Appendix C.4 Structural entropy regularization

To enforce compactness of the selected subgraph, we define:

$$L_{SE}(G^a_{sub}) = -\sum_{i,j\in V^a_{sub}} D_i D_j (1 + w^*_{ij}) \log\big(D_i D_j (1 + w^*_{ij})\big). \tag{C14}$$

where D_i is the normalized node degree and w^*_{ij} is the normalized edge weight.

This term penalizes structurally diffuse subgraphs and encourages concentrated, informative conformer selection.

Appendix D More experimental results

Table D1 Comparative performance of various methods in qualitative and quantitative interactive tasks. The best-performing methods are highlighted in bold, while the second-best methods are underscored for emphasis.

Domains	Datasets	Metrics	Baselines					Ours
			Galactica	Chem T5	MolCA	MolT5	MolTC	Gelm
DDI	TWOSIDES	ACC (↑)	$81.95_{\pm 1.72}$	$84.12_{\pm 2.41}$	$90.08_{\pm 1.89}$	$92.51_{\pm 1.72}$	$\underline{97.55}_{\pm 0.50}$	$\mathbf{98.29}_{\pm 0.36}$
		AUCROC (↑)	$87.89_{\pm 2.28}$	$89.32_{\pm 1.58}$	$93.75_{\pm 0.76}$	$93.91_{\pm 0.63}$	$\underline{98.09}_{\pm 0.51}$	$\mathbf{98.78}_{\pm 0.37}$
SSI	MNSol	MAE (↓)	$0.592_{\pm 0.093}$	$0.518_{\pm 0.040}$	$0.497_{\pm 0.056}$	$0.452_{\pm 0.078}$	$\underline{0.358}_{\pm 0.025}$	$\mathbf{0.328}_{\pm 0.015}$
		RMSE (↓)	$0.999_{\pm 0.098}$	$0.979_{\pm 0.077}$	$0.940_{\pm 0.065}$	$0.865_{\pm 0.072}$	$\underline{0.648}_{\pm 0.036}$	$\mathbf{0.594}_{\pm 0.032}$
CSI	Absorption	RMSE (↓)	$43.05_{\pm 1.42}$	$38.75_{\pm 1.86}$	$36.50_{\pm 2.07}$	$38.52_{\pm 2.24}$	$\underline{30.20}_{\pm 2.22}$	$\mathbf{29.15}_{\pm 1.82}$
	Emission	RMSE (↓)	$50.22_{\pm 2.50}$	$47.14_{\pm 2.32}$	$44.41_{\pm 1.98}$	$45.12_{\pm 1.69}$	$\underline{38.48}_{\pm 1.90}$	$\mathbf{36.07}_{\pm 1.58}$
	Lifetime	RMSE (↓)	$1.990_{\pm 0.123}$	$1.647_{\pm 0.073}$	$1.512_{\pm 0.094}$	$1.398_{\pm 0.149}$	$\underline{1.328}_{\pm 0.074}$	$\mathbf{1.210}_{\pm 0.052}$

Table D2 Comparative performance of various methods in CombiSolv-QM. The best-performing methods are highlighted in bold, while the second-best methods are underscored for emphasis.

Baseline Model		CombiSolv-QM	
		MAE (↓)	RMSE (↓)
GNN Based	MMGNN	$0.080_{\pm 0.002}$	$0.157_{\pm 0.004}$
	D-MPNN	$0.121_{\pm 0.006}$	$0.217_{\pm 0.005}$
	GEM	$\underline{0.077}_{\pm 0.003}$	$0.162_{\pm 0.002}$
	CGIB	$0.081_{\pm 0.004}$	$0.154_{\pm 0.005}$
ML Based	GOVER	$0.099_{\pm 0.004}$	$0.286_{\pm 0.007}$
	SolvBert	$0.106_{\pm 0.006}$	$0.320_{\pm 0.007}$
	Uni-Mol	$0.092_{\pm 0.006}$	$0.216_{\pm 0.005}$
	SMD	$0.105_{\pm 0.014}$	$0.342_{\pm 0.013}$
LLM Based	Galactica	$0.308_{\pm 0.014}$	$0.598_{\pm 0.018}$
	Chem T5	$0.330_{\pm 0.016}$	$0.569_{\pm 0.018}$
	MolCA	$0.304_{\pm 0.024}$	$0.553_{\pm 0.017}$
	MolT5	$0.218_{\pm 0.004}$	$0.347_{\pm 0.009}$
	MolTC	$0.086_{\pm 0.005}$	$\underline{0.151}_{\pm 0.007}$
Gelm (Ours)		$\mathbf{0.074}_{\pm 0.002}$	$\mathbf{0.143}_{\pm 0.003}$

Table D1 presents additional experimental results not included in the main text. It is worth noting that the three datasets in the CSI domain are derived from splits of the Chromophore dataset. Due to the convergence challenges faced by some DL-based baselines under these constraints, we only report the performance of the LLM-based baselines. Additionally, considering that the SSI tasks were initially fine-tuned on the CombiSolv-QM dataset, we provide the complete results for this dataset in Table D2. The observations from Table D1 and Table D2 align closely with those in the main experimental section, demonstrating that our Gelm consistently achieves superior performance over the LLM-based baseline methods across all tasks.

Appendix D.1 Further performance analysis

Time Efficiency Analysis. As efficiency has become an increasingly critical issue in current LLM systems, we conduct a time efficiency analysis of Gelm to better assist readers in evaluating its practicality and making informed choices. Specifically, we adopt the number of samples processed per unit time on a single A800 GPU as the efficiency metric. The results are shown in the Table D3. From the data in Table D3, we can observe that using a larger model as the backbone reduces the number of samples processed per unit time. Compared to MolTC, we adopt a more lightweight alignment operation, which substantially improves the model's inference efficiency. Meanwhile, we also report the average training time per epoch on 20,000 samples. Here, we adopt the average time required for each GPU to process a single sample independently. Since we adopt the LoRA fine-tuning approach, we additionally report the number of trainable parameters under different configurations to help readers better understand the training cost. We also report the GPU memory usage during training, under the same configuration as described above.

Table D3 Time Efficiency Analysis. *with Model* denotes the time efficiency after replacing the backbone.

Model	Gelm	with DeepSeek-7B	with DeepSeek-14B	with LLaMA-13B	MolTC
Throughput	3.04 it/s	2.42 it/s	2.06 it/s	2.18 it/s	2.36 it/s
GPU hours	1.80 hours	2.30 hours	2.72 hours	2.56 hours	2.29 hours
Trainable Parameters	54.9 M	74.5 M	92.9 M	85.2 M	106.9 M
GPU memory	5.4 GB	15.2 GB	29.8 GB	27.8 GB	6.2 GB

Significance Analysis. From the experimental results in the main text, we observe that Gelm achieves substantial improvements compared to the SOTA GNN-based and ML-based models. Furthermore, in comparison with the LLM-based SOTA model MolTC, we conducted additional significance analysis to validate the statistical significance of our model's performance. From Table D4, it can be observed that our improvements over MolTC are significant across all three types of datasets. Considering the model's time efficiency and training cost shown in Table D3, our Gelm model demonstrates clear superiority.

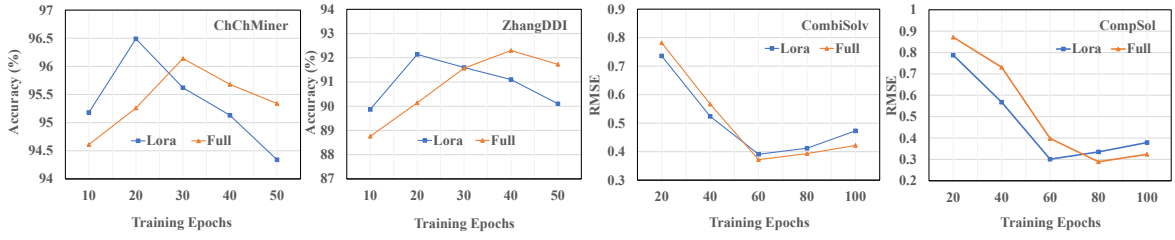
Table D4 Significance Analysis.

Model	DDI	SSI	CSI
p-value	7.89E-03	3.43E-02	9.88E-03

Although the performance improvements on certain datasets may appear modest, Gelm consistently demonstrates clear advantages. First, cross-task consistency: even when the margins are small, Gelm outperforms the baselines across diverse datasets and tasks, indicating that the improvements are systematic rather than incidental. Second, practical significance: on competitive benchmarks with strong baselines, an improvement of only 0.1–0.2 AUC can still be practically meaningful, particularly in biomedical applications where such gains may translate into substantial impact on downstream decision-making. Third, computational efficiency: compared to MolTC, which is the closest competitor in terms of performance, Gelm achieves superior efficiency, effectively reducing computational cost while maintaining SOTA accuracy.

Appendix D.2 Additional fine-tuning results

Since full-parameter fine-tuning typically requires significantly more time and GPU resources, most recent works—including MolTC [16]—adopt LoRA-based fine-tuning. To ensure fairness, our main experiments also follow this setting and apply LoRA-based adaptation. However, to further validate the effectiveness of Gelm, we additionally provide the full-parameter fine-tuning results of the best-performing model. The results are shown in the Figure D1. LoRA fine-tuning and full-parameter fine-tuning do not show significant differences in terms of the number



(a) DDI Tasks (b) SSI Tasks

Figure D1 Performance variation curves of models with full-parameter fine-tuning and LoRA fine-tuning across different datasets.

of samples processed per unit time. However, full-parameter fine-tuning usually requires a longer training time to reach optimal performance, whereas LoRA fine-tuning typically converges with fewer epochs. Under limited computational resources, LoRA fine-tuning is therefore a practical choice. Nevertheless, once full-parameter fine-tuning reaches its optimal performance, it generally achieves better results than LoRA fine-tuning. Moreover, with increasing training epochs, the performance degradation caused by overfitting tends to be less volatile compared to LoRA fine-tuning.

Appendix D.3 Protein molecule testing

In our previous evaluations, we primarily focused on small molecules, leaving the performance of LLMs on larger biomolecules, such as proteins, largely unexplored. In protein-related tasks, conformational variability plays a more critical and complex role, making accurate modeling particularly challenging. To broaden the scope of our assessment to more comprehensive biochemical and drug discovery scenarios, we conducted additional experiments on protein molecules. Due to inherent methodological limitations in existing LLM-based approaches, comparisons in this setting were limited to non-LLM methods.

Specifically, we adopted the same processed datasets as the MolTrans framework, DAVIS and BIOSNAP, for evaluating drug–target interactions (DTI) [17]. Since Gelm requires multiple conformations to capture structural variability, we augmented these datasets with 3D protein structures generated using AlphaFold2, enabling the use of 3D graph-based protein encoders. BIOSNAP comprises 4,510 drugs and 2,181 protein targets, yielding a total of 13,741 DTI pairs. It contains only positive DTI pairs, and negative pairs were generated by sampling from unseen drug–protein combinations, ensuring a balanced dataset with equal numbers of positive and negative samples. DAVIS includes Kd values for interactions among 68 drugs and 379 proteins [18], where pairs with Kd < 30 are labeled as positive. To maintain balanced training, an equal number of negative DTI pairs were incorporated.

The experimental results, summarized in Table D5, demonstrate the effectiveness of our approach in handling larger biomolecules, highlighting Gelm's ability to leverage multiple conformations and 3D structural information for accurate DTI prediction.

Table D5 Performance Metrics on Davis and BIOSNAP Datasets

Method	ROC-AUC (Davis)	PR-AUC (Davis)	ROC-AUC (BIOSNAP)	PR-AUC (BIOSNAP)
LR [19]	0.833 ± 0.011	0.235 ± 0.022	0.845 ± 0.005	0.849 ± 0.012
DNN [17]	0.865 ± 0.010	0.260 ± 0.025	0.850 ± 0.004	0.854 ± 0.009
GNN-CPI [20]	0.841 ± 0.013	0.267 ± 0.021	0.880 ± 0.006	0.891 ± 0.005
DeepDTI [21]	0.860 ± 0.003	0.233 ± 0.007	0.875 ± 0.005	0.877 ± 0.005
DeepDTA [22]	0.879 ± 0.008	0.300 ± 0.042	0.877 ± 0.006	0.884 ± 0.007
DeepConv-DTI [23]	0.885 ± 0.007	0.298 ± 0.038	0.882 ± 0.003	0.890 ± 0.006
MolTrans [17]	0.908 ± 0.003	0.402 ± 0.017	0.896 ± 0.003	0.902 ± 0.003
3DProt-DTA [24]	0.915 ± 0.004	0.397 ± 0.008	0.892 ± 0.005	0.900 ± 0.013
Ours (Gelm)	$\mathbf{0.928 \pm 0.003}$	$\mathbf{0.415 \pm 0.006}$	$\mathbf{0.917 \pm 0.004}$	$\mathbf{0.915 \pm 0.005}$

Due to the data processing and incremental pretraining time required for proteins, here we primarily focus on the pretraining and evaluation results on drug molecules. Nevertheless, our Gelm still achieves state-of-the-art performance, which is sufficient to demonstrate the effectiveness and generalizability of our model.

Appendix D.4 Hyperparameter analysis

In the main text, we mentioned three key hyperparameters: the top-k retrieval size in GTSG, the threshold τ in EGCS as well as the hyperparameter θ that controls the balance between the LLM's own loss and the information entropy loss we propose. Here, we provide a supplementary analysis of their impact on the model, helping readers understand how they influence performance. In addition, we further analyze the impact of different hyperparameter ratio settings between mutual information and structural entropy ($\alpha : \beta$) on model performance when $\theta = 1$. It is worth noting that the loss adjustment here is performed after first ensuring that the different losses are on the same order of magnitude.

Table D6 Hyperparameter Analysis. The best-performing methods are highlighted in bold.

Hyperparameter		Drugbank		DeepDDI		FreeSolv		CombiSolv	
		ACC (↑)	AUCROC (↑)	ACC (↑)	AUCROC (↑)	MAE (↓)	RMSE (↓)	MAE (↓)	RMSE (↓)
Top-K	1	$94.51_{\pm 0.26}$	$97.76_{\pm 0.30}$	$95.46_{\pm 0.27}$	$97.98_{\pm 0.29}$	$0.527_{\pm 0.017}$	$0.711_{\pm 0.033}$	$0.283_{\pm 0.026}$	$0.501_{\pm 0.019}$
	5	$95.67_{\pm 0.19}$	$98.32_{\pm 0.22}$	$96.56_{\pm 0.25}$	$98.42_{\pm 0.29}$	$0.511_{\pm 0.016}$	$0.697_{\pm 0.027}$	$0.234_{\pm 0.007}$	$0.446_{\pm 0.012}$
	10	$\mathbf{96.38}_{\pm 0.14}$	$\mathbf{99.10}_{\pm 0.12}$	$\mathbf{97.34}_{\pm 0.24}$	$\mathbf{99.08}_{\pm 0.32}$	$\mathbf{0.471}_{\pm 0.014}$	$\mathbf{0.655}_{\pm 0.030}$	$\mathbf{0.175}_{\pm 0.006}$	$\mathbf{0.384}_{\pm 0.013}$
	15	$95.44_{\pm 0.17}$	$97.98_{\pm 0.20}$	$96.49_{\pm 0.26}$	$98.30_{\pm 0.34}$	$0.520_{\pm 0.013}$	$0.691_{\pm 0.033}$	$0.227_{\pm 0.005}$	$0.420_{\pm 0.014}$
Threshold τ	0.3	$95.46_{\pm 0.15}$	$98.64_{\pm 0.23}$	$96.70_{\pm 0.22}$	$98.38_{\pm 0.31}$	$0.504_{\pm 0.012}$	$0.683_{\pm 0.029}$	$0.220_{\pm 0.006}$	$0.423_{\pm 0.015}$
	0.5	$95.77_{\pm 0.19}$	$98.73_{\pm 0.18}$	$96.82_{\pm 0.25}$	$98.51_{\pm 0.36}$	$0.499_{\pm 0.014}$	$0.672_{\pm 0.031}$	$0.209_{\pm 0.004}$	$0.420_{\pm 0.013}$
	0.7	$\mathbf{96.38}_{\pm 0.14}$	$\mathbf{99.10}_{\pm 0.12}$	$\mathbf{97.34}_{\pm 0.24}$	$\mathbf{99.08}_{\pm 0.32}$	$\mathbf{0.471}_{\pm 0.014}$	$\mathbf{0.655}_{\pm 0.030}$	$\mathbf{0.175}_{\pm 0.006}$	$\mathbf{0.384}_{\pm 0.013}$
	0.9	$94.68_{\pm 0.16}$	$97.87_{\pm 0.21}$	$95.21_{\pm 0.27}$	$97.63_{\pm 0.33}$	$0.566_{\pm 0.011}$	$0.743_{\pm 0.034}$	$0.289_{\pm 0.005}$	$0.512_{\pm 0.016}$
Trade-off θ	0.1	$94.77_{\pm 0.23}$	$98.12_{\pm 0.33}$	$95.90_{\pm 0.30}$	$98.28_{\pm 0.31}$	$0.516_{\pm 0.019}$	$0.692_{\pm 0.030}$	$0.261_{\pm 0.028}$	$0.482_{\pm 0.021}$
	0.5	$95.88_{\pm 0.21}$	$98.63_{\pm 0.20}$	$96.70_{\pm 0.23}$	$98.69_{\pm 0.31}$	$0.490_{\pm 0.018}$	$0.671_{\pm 0.030}$	$0.231_{\pm 0.009}$	$0.422_{\pm 0.015}$
	1	$\mathbf{96.38}_{\pm 0.14}$	$\mathbf{99.10}_{\pm 0.12}$	$\mathbf{97.34}_{\pm 0.24}$	$\mathbf{99.08}_{\pm 0.32}$	$\mathbf{0.471}_{\pm 0.014}$	$\mathbf{0.655}_{\pm 0.030}$	$\mathbf{0.175}_{\pm 0.006}$	$\mathbf{0.384}_{\pm 0.013}$
	2	$96.11_{\pm 0.19}$	$98.89_{\pm 0.18}$	$96.92_{\pm 0.24}$	$98.73_{\pm 0.32}$	$0.485_{\pm 0.015}$	$0.664_{\pm 0.030}$	$0.201_{\pm 0.007}$	$0.407_{\pm 0.012}$
$\alpha : \beta$	0.5	$95.55_{\pm 0.25}$	$98.73_{\pm 0.31}$	$96.33_{\pm 0.33}$	$98.58_{\pm 0.28}$	$0.501_{\pm 0.021}$	$0.684_{\pm 0.027}$	$0.242_{\pm 0.031}$	$0.453_{\pm 0.024}$
	1	$95.79_{\pm 0.22}$	$98.82_{\pm 0.18}$	$96.75_{\pm 0.25}$	$98.77_{\pm 0.29}$	$0.492_{\pm 0.020}$	$0.673_{\pm 0.028}$	$0.211_{\pm 0.012}$	$0.416_{\pm 0.017}$
	2	$\mathbf{96.38}_{\pm 0.14}$	$\mathbf{99.10}_{\pm 0.12}$	$\mathbf{97.34}_{\pm 0.24}$	$\mathbf{99.08}_{\pm 0.32}$	$\mathbf{0.471}_{\pm 0.014}$	$\mathbf{0.655}_{\pm 0.030}$	$\mathbf{0.175}_{\pm 0.006}$	$\mathbf{0.384}_{\pm 0.013}$
	5	$95.80_{\pm 0.21}$	$98.84_{\pm 0.20}$	$97.02_{\pm 0.22}$	$98.93_{\pm 0.27}$	$0.496_{\pm 0.017}$	$0.680_{\pm 0.028}$	$0.223_{\pm 0.010}$	$0.422_{\pm 0.014}$

From Table D6, we observe that the best performance in GTSG is achieved when the top-k parameter is set to 10. In fact, according to our algorithm, if k is too small, each molecule is connected to only a few neighbors, resulting in a sparse and fragmented graph. In this case, adjacent molecules along a path tend to have higher similarity because the edges are strong connections, but the graph is more likely to break apart, reducing global connectivity and limiting the LLM's ability to fully capture molecular structures during pretraining. Conversely, when k is too large, larger groups can be formed, but weakly similar edges are also introduced. This may cause dissimilar molecules to be forced into the same path, thereby diluting intra-group coherence. For this reason, we set top-k to 10 in our experiments.

For the threshold τ in EGCS, this parameter controls the construction of conformational transition relationships. When τ is set too low, the model connects almost all conformations, treating them as mutually convertible. Since molecular conformations inherently share a certain degree of similarity, our experiments show that the performance difference between $\tau = 0.5$ and $\tau = 0.3$ is already negligible. This indicates that once τ falls below a certain range, further decreasing it has little effect on model performance, which is not desirable. In contrast, when τ is set too high, model performance drops significantly. This is because each conformation is treated as an isolated node, preventing the model from capturing conformational transitions and leveraging these relationships to better understand structural variations. Therefore, in our experiments, we set $\tau = 0.7$.

Regarding the trade-off hyperparameter θ, our experiments aim to control it so that the proposed module exerts a comparable influence on the model. We observe that when θ is set such that the ratio between the module's loss and the LLM's own loss is 1, the model achieves its best performance. Conversely, reducing the influence of our module on the model leads to a rapid decline in performance, which validates the effectiveness of the proposed module. Similarly, increasing the module's influence beyond an appropriate level also causes a certain degree of performance degradation, as it excessively interferes with the model's own fine-tuning. Therefore, in our experiments, we set the balance parameter θ to 1.

For the internal ratio of mutual information to structural entropy ($\alpha : \beta$), we observed that the model achieved the best performance when this ratio was set to 2:1. When the structural loss dominates, the model tends to oversimplify the constructed graph, causing the resulting conformational graph to lose critical information. Conversely, when mutual information dominates, the conformational graph becomes overly complex, reducing the discriminability of the aggregated information. Our experiments therefore indicate that a 2:1 ratio provides the most favorable balance.

Appendix E Limitations

LLMs have made significant progress in MRL. In our study, we first followed the trend of existing LLM-based MRL models by focusing on small molecule relationships and conducted performance comparisons accordingly. We also evaluated the models on larger biomolecules, such as proteins; however, due to the inherent limitations of other LLM frameworks, we were unable to perform further comparative evaluations on large-molecule datasets.

References

1 Anonymous. Iterative substructure extraction for molecular relational learning with interactive graph information bottleneck, 2025. Manuscript submitted for publication
2 Deac A, Huang Y H, Veličković P, et al. Drug-drug adverse effect prediction with graph co-attention. arXiv preprint arXiv:1905.00534, 2019
3 Ryu J Y, Kim H U, Lee S Y. Deep learning improves prediction of drug–drug and drug–food interactions. Proceedings of the national academy of sciences, 2018, 115: E4304–E4311
4 Nyamabo A K, Yu H, Shi J Y. Ssi–ddi: substructure–substructure interactions for drug–drug interaction prediction. Briefings in Bioinformatics, 2021, 22: bbab133
5 Lee N, Hyun D, Na G S, et al. Conditional graph information bottleneck for molecular relational learning. In: ICML. PMLR2023, *Proceedings of Machine Learning Research*, volume 202. 18852–18871
6 Lee N, Yoon K, Na G S, et al. Shift-robust molecular relational learning with causal substructure. arXiv preprint arXiv:2305.18451, 2023
7 Lin S, Wang Y, Zhang L, et al. Mdf-sa-ddi: predicting drug–drug interaction events based on multi-source drug fusion, multi-source feature fusion and transformer self-attention mechanism. Briefings in Bioinformatics, 2022, 23: bbab421
8 Li Z, Zhu S, Shao B, et al. Dsn-ddi: an accurate and generalized framework for drug–drug interaction prediction by dual-view representation learning. Briefings in Bioinformatics, 2023, 24: bbac597
9 Vermeire F H, Green W H. Transfer learning for solvation free energies: From quantum chemistry to experiments. Chemical Engineering Journal, 2021, 418: 129307
10 Yu J, Zhang C, Cheng Y, et al. Solvbert for solvation free energy and solubility prediction: a demonstration of an nlp model for predicting the properties of molecular complexes. Digital Discovery, 2023, 2: 409–421
11 Meng F, Zhang H, Collins-Ramirez J S, et al. Something for nothing: Improved solvation free energy prediction with learning. 2023
12 Du W, Zhang S, Wu D, et al. 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, IJCAI 2024, Jeju, South Korea, August 3-9, 2024. ijcai.org2024. 5808–5816. URL https://www.ijcai.org/proceedings/2024/642
13 Fang X, Liu L, Lei J, et al. Geometry-enhanced molecular representation learning for property prediction. Nature Machine Intelligence, 2022, 4: 127–134
14 Rong Y, Bian Y, Xu T, et al. Self-supervised graph transformer on large-scale molecular data. In: NeurIPS, 2020
15 Zhou G, Gao Z, Ding Q, et al. Uni-mol: a universal 3d molecular representation learning framework. 2023
16 Fang J, Zhang S, Wu C, et al. Moltc: Towards molecular relational modeling in language models. arXiv preprint arXiv:2402.03781, 2024
17 Huang K, Xiao C, Glass L M, et al. Moltrans: Molecular interaction transformer for drug–target interaction prediction. Bioinformatics, 2020, 37: 830–836
18 Davis M I, Hunt J P, Herrgard S, et al. Comprehensive analysis of kinase inhibitor selectivity. Nature Biotechnology, 2011, 29: 1046–1051
19 Cao D S, Xu Q S, Liang Y Z. Propy: A tool to generate various modes of chou's pseaac. Bioinformatics, 2013, 29: 960–962
20 Tsubaki M, Tomii K, Sese J. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics, 2018, 35: 309–318
21 Wen M, et al. Deep-learning-based drug–target interaction prediction. Journal of Proteome Research, 2017, 16: 1401–1409
22 Öztürk H, Özgür A, Ozkirimli E. Deepdta: Deep drug–target binding affinity prediction. Bioinformatics, 2018, 34: i821–i829
23 Lee I, Keum J, Nam H. Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences. PLOS Computational Biology, 2019, 15
24 Voitsitskyi T, Stratiichuk R, Koleiev I, et al. 3dprotdta: A deep learning model for drug-target affinity prediction based on residue-level protein graphs. RSC Advances, 2023, 13: 10261–10272