
ModuLM: Enabling Modular and Multimodal Molecular Relational Learning with Large Language Models

Zhuo Chen^{1,2}, Yizhen Zheng⁴, Huan Yee Koh⁴, Hongxin Xiang³, Linjiang Chen⁵,
Wenjie Du^{1,2,5,†}, Yang Wang^{1,2,5,†}

¹University of Science and Technology of China, China

²Suzhou Institute for Advanced Research, USTC, China

³Hunan University, China ⁴Monash University, Australia

⁵State Key Laboratory of Precision and Intelligent Chemistry, USTC, China

{czchenzhuo, duwenjie}@mail.ustc.edu.cn

{yizhen.zhengl, huan.koh}@monash.edu

xianghx@hnu.edu.cn {linjiangchen, angyan}@ustc.edu.cn

Abstract

Molecular Relational Learning (MRL) aims to understand interactions between molecular pairs, playing a critical role in advancing biochemical research. With the recent development of large language models (LLMs), a growing number of studies have explored the integration of MRL with LLMs and achieved promising results. However, the increasing availability of diverse LLMs and molecular structure encoders has significantly expanded the model space, presenting major challenges for benchmarking. Currently, there is no LLM framework that supports both flexible molecular input formats and dynamic architectural switching. To address these challenges, reduce redundant coding, and ensure fair model comparison, we propose ModuLM, a framework designed to support flexible LLM-based model construction and diverse molecular representations. ModuLM provides a rich suite of modular components, including 8 types of 2D molecular graph encoders, 11 types of 3D molecular conformation encoders, 7 types of interaction layers, and 7 mainstream LLM backbones. Owing to its highly flexible model assembly mechanism, ModuLM enables the dynamic construction of over 50,000 distinct model configurations. In addition, we provide comprehensive results to demonstrate the effectiveness of ModuLM in supporting LLM-based MRL tasks. ModuLM is available at <https://github.com/ssjjjhw/ModuLM>.

1 Introduction

Molecular Relational Learning (MRL) [36], which aims to understand the interactions between molecular pairs, has garnered growing attention due to its wide-ranging applications across various scientific domains [55]. For example, drug-drug interactions (DDIs) are vital for understanding the effects of concurrent drug use, which can inform strategies to prevent adverse drug reactions and ensure patient safety [42], while solute-solvent interactions (SSIs) are fundamental to solution chemistry and are pivotal in the design and optimization of chemical processes [66, 8, 10]. However, the exhaustive experimental validation of these interactions is notoriously time-consuming and costly.

In recent years, large language models (LLMs) have emerged as a promising new paradigm in MRL research due to their powerful capabilities in knowledge integration and reasoning. Compared with

† : corresponding author

traditional methods, LLMs can more efficiently process and understand complex interactions between molecules, significantly improving modeling performance and generalizability. A growing body of research has focused on LLM-based MRL frameworks [52, 29, 17], leveraging the strengths of LLMs to achieve strong results on MRL tasks. For instance, ReactionT5[57] proposed a text-based pretrained LLM tailored for MRL tasks, while MolTC[17] further advanced this line of research by integrating multimodal data and incorporating 2D molecular graphs for improved performance. These developments highlight the research value and application potential of LLMs in MRL. However, with the emergence of an increasing number of encoding methods and backbone models [51, 28, 70, 87, 76, 64, 65, 24], it is now possible to adopt more flexible strategies to recombine components and build more novel model architectures. However, this flexibility introduces new challenges for the benchmarking and evaluation of LLM-based MRL models.

Lack of diverse input support: Molecular structures can typically be represented in various forms, such as 1D SMILES strings, 2D molecular graphs, and 3D molecular conformations; however, most existing models support only a single representation modality, commonly 1D SMILES[33, 22, 75] or 2D graphs[20, 59, 9, 12], which limits their ability to fully capture the complexity of molecular interactions and may result in the loss of critical structural information. In the field of LLMs, MRL models that accept 3D molecular structure inputs remain extremely rare, despite the fact that specific 3D conformations are often essential for accurately modeling chemical phenomena, for instance, in small-molecule binding to target proteins[72]. These issues highlight the importance of a unified framework that can accommodate 1D, 2D, and 3D molecular inputs to enable more comprehensive, flexible, and accurate molecular relational learning across a wide range of task scenarios.

Lack of Flexible Architectures: Current LLM-based MRL models often adopt relatively rigid architectures. For example, ReactionT5 [57] uses a unified model to encode both SMILES sequences and molecular property descriptions, while MolTC [17] employs graph neural networks (GNNs) to encode molecular graphs. Although these methods achieve performance improvements, they are still somewhat limited by their encoding strategies. In the non-LLM MRL domain, many more effective encoding strategies have been developed, such as MMGNN [11], which employs interpretable GNNs to extract key subgraphs for solvation free energy prediction, and Uni-Mol [84], which introduces pre-trained SE(3) Transformer models specifically designed for molecular data. However, integrating these encoding methods into existing LLM frameworks remains a challenge. Furthermore, most current LLM-based models overlook the modeling of molecular interaction features. These challenges highlight the importance of developing a model framework that can seamlessly combine diverse encoding modules while effectively incorporating molecular interaction information.

To this end, we propose **ModuLM**, a unified and extensible framework designed to overcome the limitations of existing LLM-based MRL approaches. ModuLM provides a highly flexible model construction mechanism that supports a wide range of molecular input formats, multimodal integration strategies, and diverse prompt designs. The framework accepts molecular representations in the form of 1D SMILES, 2D molecular graphs and 3D conformations. It includes 8 types of 2D molecular graph encoders, 11 types of molecular conformation encoders, 7 types of interaction feature encoders, and 7 mainstream LLM backbones, along with specially designed prompt templates for integrating different types of molecular features. To enhance usability and extensibility, ModuLM adopts a modular interface design that allows users to flexibly assemble and extend models, supports incremental pretraining, and is capable of handling complex molecular interaction modeling tasks. ModuLM can generate over 50,000 distinct model configurations. We conduct comprehensive benchmark experiments on tasks such as DDI, SSI and CSI. The results demonstrate that ModuLM performs remarkably well in constructing, evaluating, and comparing LLM-based MRL models. These findings highlight ModuLM’s strong potential to advance the development of MRL models and provide valuable insights into molecular interaction mechanisms.

2 Related Work

Molecular Relational Learning: MRL is critical for drug research, with machine learning offering a scalable alternative to costly experimental validation [37]. Early methods focused on GNNs [33, 82, 78, 18, 71], such as Nyamabo et al.’s substructure-level interaction model using GAT and co-attention [83], and Lee et al.’s CGIB [36], which applies the information bottleneck to extract key substructures. LLM-based methods have gained momentum. For instance, ReactionT5 [53] enhances molecular understanding by integrating chemical structures with natural language. MolTC [17]

further advances this direction by combining 2D molecular graph features with chain-of-thought reasoning to support complex molecular inference.

LLMs in the Molecular Domain: LLMs have been widely applied in 1D, 2D, and 3D molecular pattern learning [5]. For 1D, methods like MolT5 [14] and KV-PLM [77] tokenize SMILES strings for representation learning. In 2D, approaches such as Text2Mol [16], MolCA [45], and DrugChat [40] integrate molecular graphs with text encoders or LLMs. For 3D, MolLM [62] and 3D-MoLM [39] incorporate spatial relationships via attention mechanisms and 3D encoders. In addition, LLMs have also found applications in MRL, such as ReactionT5 [57] and MolTC [17], which utilize multimodal data, including molecular graphs (2D), chemical properties, and SMILES (1D), for MRL.

Deep Learning Frameworks Specialized for MRL: DeepPurpose is a user-friendly deep learning library for drug-target interaction prediction. It supports customized model training with 15 compound and protein encoders and over 50 neural architectures [27]. FlexMol is another toolkit designed for MRL, offering a variety of encoders and interaction layers that support sequence-based and graph-based representations of drugs and proteins [61].

3 ModuLM

In this section, we introduce ModuLM following the model training workflow for LLM-based MRL.

3.1 Framework

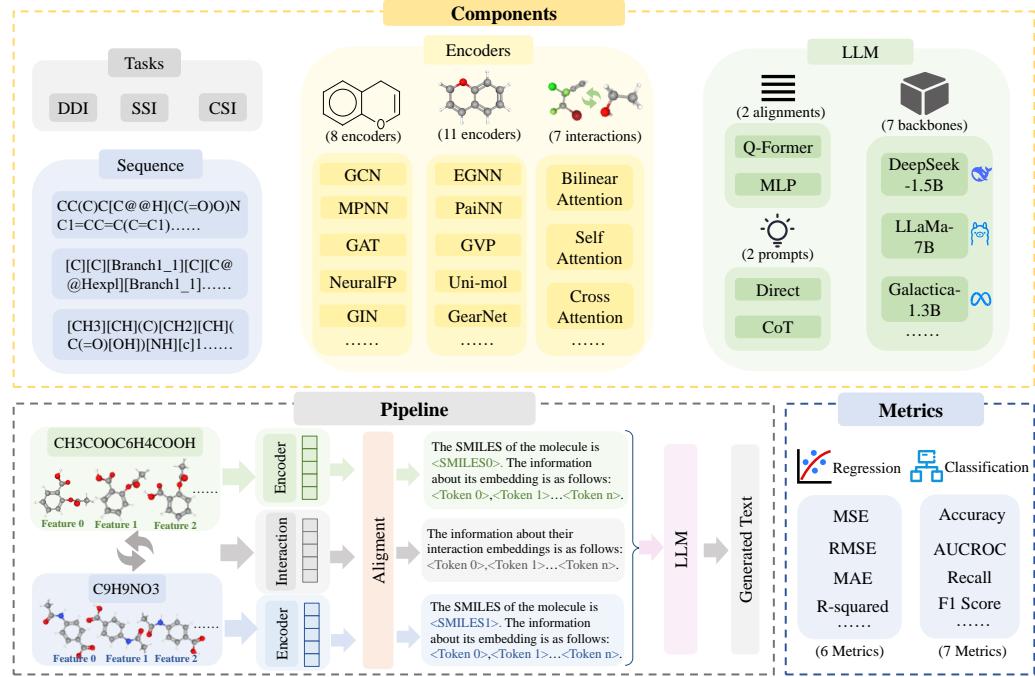


Figure 1: Overview of the ModuLM framework.

ModuLM is an LLM framework designed for MRL, with the overall architecture illustrated in Figure 1. It supports three types of molecular text inputs and accommodates a variety of molecular inputs across 1D, 2D, and 3D modalities. The framework provides eight 2D molecular graph encoders, eleven 3D molecular conformation encoders, and incorporates seven feature interaction designs. It also supports evaluation across multiple task types.

3.2 Post Pretraining

To strengthen the domain-specific capabilities of LLMs in chemistry, we begin with incremental pretraining to better adapt them for MRL tasks. In ModuLM, we conduct a survey of various

Table 1: Text settings for different pretraining methods.

Molecular Interaction-based Pretraining The first molecule has a SMILES representation of <SMILES0>, which suggests certain structural characteristics and chemical functionalities. Based on its structure, it may exhibit the property [Property0], such as high solubility, bioactivity, or specific binding affinity. On the other hand, the second molecule is represented by the SMILES string <SMILES1>, and analysis of its structure indicates it may exhibit the property [Property1], potentially contributing to its pharmacokinetic behavior or molecular interaction profile.
Substructure-based Pretraining The molecule has a SMILES representation of <SMILES0>, which encodes its atomic connectivity and overall molecular structure. It contains notable substructures such as <substructe0> and <substructe1>, both of which are known to play significant roles in determining the molecule's physicochemical and biological properties. Based on the presence of these functional groups or structural motifs, the molecule is likely to exhibit the properties [Property0] and [Property1], which may influence its reactivity, solubility, or interaction with biological targets.

authoritative biochemical databases, such as PubChem¹ and DrugBank [32], collecting a large amount of molecular property description texts. We then provide three distinct pretraining strategies: Molecular Interaction-based pretraining, Substructure-based pretraining and Structure Similarity-guided Grouping pretraining.

Molecular Interaction-based Pretraining is based on the pretraining method proposed by MolTC [17]. Its pretraining text setup is shown in the Table 1. Considering that molecular interaction tasks typically involve two different molecules, MolTC integrates the textual representations of both molecules and inputs them jointly during pretraining, enabling the LLM to develop prior knowledge in the form of molecular pairs.

Substructure-based Pretraining adopts a pretraining text setup as shown in the Table 1, enabling the LLM to learn more fine-grained information within molecules. This allows the LLM to make more accurate judgments based on the potential substructures of different molecules, thereby enhancing its generalization ability when encountering previously unseen molecules and improving performance on downstream tasks.

Structure Similarity-guided Grouping Pretraining combines substructure-based pretraining with a grouping strategy based on structural similarity. Molecules with similar structures are grouped together and input as a group during pretraining, thereby enhancing the LLM's understanding of this class of molecules. This method shares the same prompt configuration as the substructure-based pretraining approach, differing in the ordering of the input.

It is worth noting that if you use the Q-former approach for aligning textual and molecular data, additional pretraining will be required. The specific pretraining process can be selected based on your needs. Here, we provide MolTC's [17] methods for aligning molecules with text and molecular data.

3.3 Fine-tuning

3.3.1 Input Data

To transform raw molecules into meaningful representations, ModuLM first performs preprocessing followed by encoding. The preprocessing stage includes tasks such as tokenization, normalization, feature extraction, fingerprint generation, molecular graph construction, and molecular conformation generation. The encoding stage is responsible for dynamically constructing the input features for the LLM. In this stage, the preprocessed data is processed to generate embeddings that can be utilized by subsequent network layers. The following will present the specific procedures and methods used by ModuLM to encode different types of data.

1D Representations of Molecules: The commonly used molecular representations for MRL in current LLMs are typically SMILES and SelfIES [35]. However, SMILES representations may have

¹<https://pubchem.ncbi.nlm.nih.gov>

difficulties clearly expressing certain molecular structures. Therefore, for MRL tasks, we recommend using SelfIES. Nonetheless, to provide a more comprehensive benchmarking framework for previous LLM-based approaches, we include both molecular text encoding methods here, and additionally incorporate SMARTS. For 1D representations of molecules, the encoding method depends on the specific LLM used.

2D Molecular Graph Representation: Since 1D molecular representations often fail to capture the structural information of molecules, it is common in MRL tasks to incorporate multimodal molecular structure information. In non-LLM research areas, molecular graph representation is the most widely used approach. In most tasks, molecular graphs help models better understand molecular structures, thereby enhancing MRL performance. We provide a variety of methods as shown in the Table 2.

3D Molecular Conformation Representation: In most MRL tasks, 2D molecular graphs are sufficient for the model to make effective judgments. However, in certain specialized tasks, it is necessary to incorporate the 3D spatial information of molecules to accurately determine intermolecular interactions. This is an aspect that current LLM-based MRL models have largely overlooked. In ModuLM, we address this limitation by providing various 3D structure encoding methods to support more advanced molecular relation learning. The encoding methods are listed in the Table 2.

Interaction: In current LLM-based MRL model designs, the explicit modeling of molecular interaction relationships is often overlooked. In ModuLM, we address this issue by introducing specially designed prompts and interaction layers to incorporate interaction information into the LLM. The Interaction Layer is another key building block of the MRL model. These layers serve two main functions: capturing and modeling the relationships between different molecular entities, and integrating multiple embeddings of the same entity to form a more comprehensive feature representation. The Interaction Layer can accept encoded inputs from different molecules, enabling the construction of more complex model architectures. In ModuLM, the interaction designs we provide are shown in Table 2.

Table 2: Encoding methods for different data formats

Encoder Type	Methods
2D Graph	GCN [34], MPNN [22], GAT [68], NeuralFP [13], AttentiveFP [73], GIN [74], GraphSAGE[25], CoATGIN[80]
3D Conformation	EGNN [58], 3D-GeoFormer [86], SE3Transformer [19], PaiNN [59], GVP [30], GearNet [81], DimeNet++ [21], SchNet [60], SphereNet [44], G-SphereNet[46], Uni-mol [85]
Interaction	Bilinear Attention[2], Self Attention[67], Cross Attention[54], Highway[87], Gated Fusion[47], Bilinear Fusion[41], Mean

3.3.2 Alignment

Since data from different modalities typically exist in distinct semantic spaces, it is necessary to perform alignment before feeding them into the LLM. Currently, two common approaches are used: employing a lightweight MLP [43] or using a Q-former [38]. In the ModuLM framework we provide, both alignment methods are supported. It is worth noting that when using a Q-former, it is typically involved during the pretraining stage to enable better alignment performance.

3.3.3 Backbone

Existing LLM-based MRL models often adopt different backbone architectures, and there has been no systematic investigation into the MRL performance across different LLM backbones. In ModuLM, we provide a streamlined method for switching backbones. We have surveyed and integrated a range of mainstream LLMs and offer simple interfaces for replacement, enabling easier comparison of performance differences across various types and scales of LLMs under a unified experimental setup. Here, we provide two types of prompts: direct inference and chain-of-thought-based reasoning. The specific prompt designs are detailed in the Appendix A.3.5.

3.4 Evaluation Metrics

ModuLM supports multiple default metrics, aligning with the TDC standard for molecular relational learning[26]. Users can specify the metrics in the Trainer for early stopping and testing. These metrics include various regression metrics (Mean Squared Error (MSE), Root-Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), Pearson Correlation Coefficient (PCC), Spearman Correlation Coefficient), binary classification metrics (Area Under Receiver Operating Characteristic Curve (AUC-ROC), Area Under the Precision-Recall Curve (PR-AUC), Range LogAUC, Accuracy Metrics, Precision, Recall, F1 Score).

3.5 Supporting Datasets

ModuLM is compatible with all MRL datasets that conform to our specified format. These datasets typically consist of three components: molecular entity one, molecular entity two, and a label. We provide utility functions to facilitate the loading of datasets in this format. In addition, ModuLM includes a wide range of built-in datasets from various domains, such as Drugbank (Version 5.0.3), ZhangDDI [79], ChChMiner [88], DeepDDI [56], TWOSIDES [63], Chromophore [31], MNSol [48], CompSol [50], Abraham [23], CombiSolv [69], FreeSolv [49], and CombiSolv-QM [69]. For more details, please refer to the Appendix A.2.

4 Experiments

We conduct validation experiments on MRL tasks using ModuLM to demonstrate the framework’s capability in supporting a wide range of experiments, comparisons, and analyses. The following sections present the results for DDI tasks, showcasing the impact of different inputs and encoders on the performance of LLMs in MRL. More experimental details are provided in the Appendix A.5.

4.1 Experimental Setup

Table 3: Experimental Settings on DDI Datasets

Experiment No.	Backbone	Encoder	Interaction	Input Feature
1.1	Galactica-1.3B	-	-	m_s
1.2	Galactica-1.3B	GIN	-	$m_s + m_g$
1.3	Galactica-1.3B	GIN	Cross Attention	$m_s + m_g$
1.4	Galactica-1.3B	Uni-mol	-	$m_s + m_c$
1.5	Galactica-6.7B	MPNN	Gated Fusion	$m_s + m_g$
1.6	DeepSeek-1.5B	-	-	m_s
1.7	DeepSeek-1.5B	GIN	-	$m_s + m_g$
1.8	DeepSeek-1.5B	Uni-mol	-	$m_s + m_c$
1.9	DeepSeek-7B	3D-GeoFormer	Highway	$m_s + m_c$
1.10	DeepSeek-14B	Uni-mol	-	$m_s + m_c$
1.11	DeepSeek-14B	GAT	Self Attention	$m_s + m_g$
1.12	LLaMA-1B	-	-	m_s
1.13	LLaMA-1B	CoATGIN	-	$m_s + m_g$
1.14	LLaMA-1B	EGNN	Gated Fusion	$m_s + m_c$
1.15	LLaMA-13B	SchNet	Bilinear Attention	$m_s + m_c$

Note: m_s = molecular sequence, m_g = molecular graph, m_c = molecular conformation. ‘-’ indicates that no method is applied.

Given the flexibility of ModuLM, which enables a large number of potential model combinations, the goal of this section is not to exhaustively explore the entire model space. Instead, we select several model combinations as examples to demonstrate ModuLM’s robust capabilities in constructing and evaluating diverse model architectures across various datasets and performance metrics. It is worth noting that for different backbones, we adopt a unified pretraining strategy. The experimental backbones presented in the main text are LLMs pretrained based on the Structure Similarity-guided Grouping pretraining approach.

During the evaluation phase on downstream tasks, we utilized the same datasets used in the MolTC framework [17] for evaluating MRL tasks, including DrugBank (Version 5.0.3), ZhangDDI [79], ChChMiner [88], DeepDDI [56], TWOSIDES [63], Chromophore [31], MNSol [48], CompSol [50], Abraham [23], CombiSolv [69], FreeSolv [49], and CombiSolv-QM [69]. We further process these datasets using RDKit by generating multiple conformations for each molecule, aiming to explore the performance of ModuLM on 3D conformation-based LLM-driven MRL tasks. Based on ModuLM, we construct 15 models, as detailed in Table 3, and compare them with five state-of-the-art LLM-based models for MRL tasks: Galactica, ChemT5 [6], MolT5, MolCA [45], and MolTC [17].

For the DDI, SSI and CSI datasets, we randomly split the data into training, validation, and testing sets in a ratio of 7:2:1. Each experiment was repeated five times to mitigate the effects of randomness, and the average results were reported. All experiments were conducted using eight NVIDIA A100 80G GPUs. For more specific training details, please refer to the Appendix A.5.

4.2 Experimental Results and Analysis

Table 4: Performance on DDI Datasets

Experiment	AUC-ROC (ChChMiner)	Accuracy (ChChMiner)	AUC-ROC (ZhangDDI)	Accuracy (ZhangDDI)	AUC-ROC (DeepDDI)	Accuracy (DeepDDI)
Chem T5[7]	0.867 \pm 0.012	0.814 \pm 0.009	0.889 \pm 0.017	0.751 \pm 0.021	0.856 \pm 0.012	0.784 \pm 0.013
MolCA[45]	0.924 \pm 0.006	0.901 \pm 0.009	0.895 \pm 0.006	0.745 \pm 0.010	0.878 \pm 0.014	0.841 \pm 0.015
MolT5[15]	0.914 \pm 0.019	0.862 \pm 0.022	0.901 \pm 0.011	0.802 \pm 0.015	0.907 \pm 0.014	0.870 \pm 0.016
MolTC[17]	0.964 \pm 0.008	0.957 \pm 0.006	0.941 \pm 0.006	0.896 \pm 0.008	0.977 \pm 0.013	0.956 \pm 0.011
1.1	0.933 \pm 0.011	0.924 \pm 0.009	0.912 \pm 0.008	0.854 \pm 0.004	0.899 \pm 0.010	0.855 \pm 0.009
1.2	0.956 \pm 0.008	0.943 \pm 0.009	0.930 \pm 0.006	0.872 \pm 0.009	0.924 \pm 0.008	0.887 \pm 0.008
1.3	0.960 \pm 0.010	0.954 \pm 0.006	0.933 \pm 0.007	0.891 \pm 0.004	0.939 \pm 0.007	0.904 \pm 0.008
1.4	0.955 \pm 0.005	0.949 \pm 0.008	0.936 \pm 0.014	0.901 \pm 0.010	0.956 \pm 0.008	0.919 \pm 0.007
1.5	0.940 \pm 0.009	0.932 \pm 0.008	0.921 \pm 0.008	0.866 \pm 0.005	0.948 \pm 0.009	0.912 \pm 0.006
1.6	0.936 \pm 0.010	0.930 \pm 0.011	0.920 \pm 0.009	0.860 \pm 0.008	0.906 \pm 0.008	0.872 \pm 0.008
1.7	0.957 \pm 0.008	0.953 \pm 0.010	0.934 \pm 0.006	0.889 \pm 0.004	0.958 \pm 0.007	0.942 \pm 0.007
1.8	0.966 \pm 0.007	0.964 \pm 0.005	0.938 \pm 0.005	0.907 \pm 0.006	0.972 \pm 0.009	0.959 \pm 0.010
1.9	0.944 \pm 0.010	0.935 \pm 0.009	0.925 \pm 0.005	0.870 \pm 0.003	0.955 \pm 0.008	0.930 \pm 0.007
1.10	0.931 \pm 0.012	0.918 \pm 0.010	0.916 \pm 0.009	0.861 \pm 0.011	0.943 \pm 0.010	0.915 \pm 0.008
1.11	0.935 \pm 0.007	0.921 \pm 0.012	0.906 \pm 0.009	0.855 \pm 0.008	0.936 \pm 0.009	0.908 \pm 0.007
1.12	0.925 \pm 0.013	0.911 \pm 0.011	0.901 \pm 0.008	0.852 \pm 0.006	0.895 \pm 0.010	0.850 \pm 0.009
1.13	0.945 \pm 0.009	0.937 \pm 0.008	0.925 \pm 0.007	0.870 \pm 0.006	0.935 \pm 0.008	0.904 \pm 0.008
1.14	0.951 \pm 0.007	0.946 \pm 0.011	0.928 \pm 0.004	0.875 \pm 0.005	0.946 \pm 0.007	0.918 \pm 0.006
1.15	0.915 \pm 0.016	0.896 \pm 0.013	0.913 \pm 0.008	0.860 \pm 0.002	0.928 \pm 0.011	0.897 \pm 0.008

Table 5: Performance on SSI Datasets

Experiment	MAE (FreeSolv)	RMSE (FreeSolv)	MAE (CompSol)	RMSE (CompSol)	MAE (CombiSolv)	RMSE (CombiSolv)
Chem T5[7]	0.923 \pm 0.022	1.511 \pm 0.043	0.611 \pm 0.017	0.766 \pm 0.032	0.840 \pm 0.040	1.294 \pm 0.043
MolCA[45]	0.761 \pm 0.034	1.303 \pm 0.039	0.505 \pm 0.036	0.726 \pm 0.040	0.771 \pm 0.033	1.130 \pm 0.027
MolT5[15]	0.733 \pm 0.047	1.135 \pm 0.059	0.496 \pm 0.028	0.708 \pm 0.020	0.677 \pm 0.024	1.066 \pm 0.027
MolTC[17]	0.533 \pm 0.018	0.726 \pm 0.022	0.244 \pm 0.018	0.356 \pm 0.022	0.237 \pm 0.019	0.465 \pm 0.022
1.1	0.710 \pm 0.021	1.120 \pm 0.030	0.472 \pm 0.024	0.665 \pm 0.028	0.615 \pm 0.026	0.984 \pm 0.032
1.2	0.570 \pm 0.020	0.910 \pm 0.028	0.384 \pm 0.022	0.540 \pm 0.025	0.568 \pm 0.023	0.930 \pm 0.030
1.3	0.556 \pm 0.018	0.840 \pm 0.025	0.366 \pm 0.021	0.522 \pm 0.024	0.487 \pm 0.021	0.820 \pm 0.027
1.4	0.534 \pm 0.017	0.808 \pm 0.024	0.347 \pm 0.020	0.501 \pm 0.023	0.447 \pm 0.020	0.780 \pm 0.026
1.5	0.580 \pm 0.016	0.972 \pm 0.023	0.403 \pm 0.019	0.575 \pm 0.021	0.579 \pm 0.019	0.898 \pm 0.025
1.6	0.685 \pm 0.015	1.086 \pm 0.022	0.451 \pm 0.018	0.643 \pm 0.020	0.602 \pm 0.018	0.945 \pm 0.024
1.7	0.550 \pm 0.014	0.749 \pm 0.021	0.271 \pm 0.017	0.415 \pm 0.019	0.289 \pm 0.017	0.515 \pm 0.023
1.8	0.510 \pm 0.013	0.698 \pm 0.020	0.191 \pm 0.016	0.298 \pm 0.018	0.190 \pm 0.016	0.388 \pm 0.022
1.9	0.555 \pm 0.014	0.825 \pm 0.021	0.335 \pm 0.017	0.490 \pm 0.019	0.393 \pm 0.017	0.595 \pm 0.023
1.10	0.605 \pm 0.015	0.850 \pm 0.022	0.443 \pm 0.018	0.598 \pm 0.020	0.548 \pm 0.018	0.864 \pm 0.024
1.11	0.590 \pm 0.016	0.876 \pm 0.023	0.458 \pm 0.019	0.601 \pm 0.021	0.556 \pm 0.019	0.839 \pm 0.025
1.12	0.745 \pm 0.017	1.091 \pm 0.024	0.514 \pm 0.020	0.692 \pm 0.022	0.687 \pm 0.020	1.008 \pm 0.026
1.13	0.605 \pm 0.016	0.880 \pm 0.023	0.374 \pm 0.019	0.530 \pm 0.021	0.538 \pm 0.019	0.820 \pm 0.025
1.14	0.580 \pm 0.015	0.887 \pm 0.022	0.360 \pm 0.018	0.515 \pm 0.020	0.460 \pm 0.018	0.790 \pm 0.024
1.15	0.630 \pm 0.018	0.910 \pm 0.025	0.450 \pm 0.021	0.633 \pm 0.023	0.567 \pm 0.021	0.892 \pm 0.027

Table 4 and Table 5 show the experimental results of commonly used MRL LLMs, as well as their performance under our experimental settings. It is worth noting that in our experimental setup, we adopt a direct inference approach using LLMs without employing chain-of-thought reasoning. In our setup, we design models with different backbones, input formats, and encoders. Due to space limitations, additional experimental results are presented in the Appendix A.5.6.

Impact of Input Data and Encoders: We use the settings {1.1, 1.2, 1.4}, {1.6, 1.7, 1.8}, and {1.12, 1.13} to evaluate ModuLM’s ability to analyze and integrate different inputs and encoder layers within LLMs. The results in the table indicate that integrating multimodal information, such as 2D molecular graphs and 3D molecular conformations, can indeed enhance model performance. Notably, models that incorporate 3D molecular conformation information achieve the best results.

Impact of Interaction Layers: We use the configurations {1.3, 1.5, 1.9, 1.11, 1.14, 1.15} to evaluate ModuLM’s ability to analyze the effect of feature interaction layers. We first conduct experiments with various non-interaction designs. The analysis shows that adding interaction layers consistently improves model performance to some extent. This confirms the importance of interaction layers in LLM-based MRL models. However, existing LLM comparisons generally ignore multi-molecular interaction information. ModuLM enables multi-dimensional analysis, which helps better assess the impact of different types of multimodal information on model performance.

Impact of Different Backbones: Existing LLM-based MRL tasks generally lack systematic evaluation across different backbones. However, thoroughly testing LLMs under various experimental configurations requires significant time and resources. ModuLM offers a framework for efficient and rapid evaluation. As shown in Table 4 and Table 5, we conduct experiments using different backbones. Among them, models from the DeepSeek series often achieve better performance. Interestingly, our results reveal that larger model sizes do not necessarily lead to better performance in MRL tasks. This may be because larger LLMs possess stronger generalization ability, which can limit task-specific adaptation during fine-tuning. In contrast, smaller LLMs adapt better during fine-tuning, leading to stronger task specialization in MRL scenarios.

4.3 Custom Model Design and Evaluation

This section demonstrates how users can leverage ModuLM to extend, construct, and analyze more complex models. Figure 2 presents an overview of the newly proposed model architecture. This

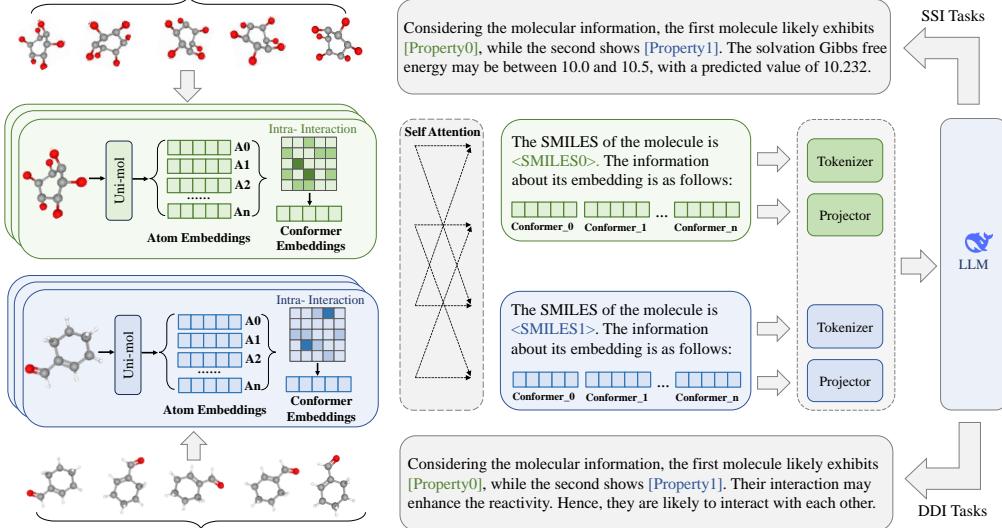


Figure 2: An overview of the custom model designed using ModuLM.

design builds upon the best-performing configuration identified in Experiment 1.8, which utilizes 3D molecular conformations, employs Uni-Mol as the encoder, and adopts DeepSeek-1.5B as the LLM backbone. In this enhanced version, we further refined the encoder and the dual-molecule interaction mechanism, and incorporated CoT prompting strategy to boost the model’s reasoning capabilities.

To simulate practical scenarios in which users may wish to combine predefined encoders with custom modules, we reimplement the molecular encoder as a user-defined component and integrate it with other existing encoder modules to construct a complete, functional model. Specifically, we redesign the representation of atom-level data encoded by Uni-Mol by calculating the attention weights of each

atom to all other atoms within the conformation. These weights are then used to reaggregate the atom features, and the mean of the weighted features serves as the new conformation-level representation. The full model construction process based on ModuLM is detailed in Appendix A.4.

We retained the experimental settings described in Section 4.1 and conducted ablation studies to reflect real-world use cases of ModuLM. In these experiments, *w/o M-Encoder* refers to the exclusion of the customized encoder, *w/o Interaction* indicates the removal of the molecule interaction design, and *w/o CoT* represents the absence of the chain-of-thought reasoning mechanism. The full experimental results are reported in Table 6, while the outcomes of the ablation studies are summarized in Table 7.

Table 6: Performance of Custom Model on DDI and SSI Datasets

Experiment	Accuracy (ChChMiner)	Accuracy (ZhangDDI)	Accuracy (DeepDDI)	RMSE (FreeSolv)	RMSE (CompSol)	RMSE (CombiSolv)
1.7	0.953 ± 0.010	0.889 ± 0.004	0.942 ± 0.007	0.749 ± 0.021	0.415 ± 0.019	0.515 ± 0.023
1.8	0.964 ± 0.005	0.907 ± 0.006	0.959 ± 0.010	0.698 ± 0.020	0.298 ± 0.018	0.388 ± 0.022
Custom Model	0.968 ± 0.006	0.911 ± 0.006	0.964 ± 0.008	0.680 ± 0.019	0.288 ± 0.013	0.359 ± 0.013

Table 7: Results of the Ablation Study on DDI and SSI Datasets

Experiment	Accuracy (ChChMiner)	Accuracy (ZhangDDI)	Accuracy (DeepDDI)	RMSE (FreeSolv)	RMSE (CompSol)	RMSE (CombiSolv)
w/o M-Encoder	0.962 ± 0.007	0.905 ± 0.005	0.959 ± 0.007	0.700 ± 0.019	0.299 ± 0.017	0.370 ± 0.020
w/o Interaction	0.955 ± 0.008	0.896 ± 0.006	0.950 ± 0.007	0.705 ± 0.018	0.317 ± 0.016	0.388 ± 0.019
w/o CoT	0.959 ± 0.006	0.901 ± 0.006	0.956 ± 0.008	0.732 ± 0.020	0.335 ± 0.018	0.382 ± 0.021
Full Model	0.968 ± 0.006	0.911 ± 0.006	0.964 ± 0.008	0.680 ± 0.019	0.288 ± 0.013	0.359 ± 0.013

Through the above experiments, we validated ModuLM’s strong capability in supporting user-defined models. Users can define custom encoders following our provided protocol and flexibly integrate them with other encoders and interaction layers as modular components. The customized model involves a more complex configuration, including user-defined blocks and additional operations such as stacking and flattening the outputs of ModuLM components. Moreover, with ModuLM’s dynamic model construction mechanism, users can easily adjust the model structure to perform ablation studies. As shown in Table 6 both the CoT reasoning prompt and the interaction layer contribute to performance improvements. Additionally, our custom encoder module, which incorporates internal interaction mechanisms, also enhances model performance.

5 Conclusion

We propose ModuLM, a framework that supports the dynamic construction of LLM-based MRL models to address benchmarking challenges in LLM-driven molecular relational learning. ModuLM accommodates multiple molecular input formats and enables the flexible assembly of diverse model architectures, facilitating robust and scalable experimentation. The framework simplifies model development and standardizes the evaluation process across different architectures, ensuring fair and consistent benchmarking. By providing a flexible and modular platform, ModuLM not only advances research in the MRL field but also lays the foundation for cross-disciplinary collaboration and innovation, with broad applications in areas such as drug design and molecular interaction analysis, helping researchers accelerate critical biomedical discoveries.

Limitations: The benchmark experiments presented in this paper demonstrate the capabilities of the ModuLM framework in constructing and comparing various model architectures, yet they do not cover all possible model combinations. Our primary goal is to highlight ModuLM’s advantages in enabling flexible model construction and efficient model comparison, showcasing its adaptability and effectiveness across diverse model architectures. While the current experiments provide valuable insights into the functionality and potential of the framework, a comprehensive exploration of all possible model combinations is beyond the scope of this study and is intended as a key direction for future research. We encourage the research community to further investigate the performance of different configurations using our framework, thus promoting the diversity and innovation of model architectures.

Future Work: In the future, we will continue to maintain and expand the components within the ModuLM framework to further enhance its flexibility and applicability. Specifically, we will focus on introducing additional types of encoders, interaction layers, and evaluation metrics, while also

expanding support for a broader range of LLMs to improve the generalizability and scalability of the framework. At the same time, we will deepen ModuLM’s application in molecular relational learning tasks, particularly in areas such as drug discovery and protein interaction analysis. We expect that through continuous iteration and optimization, ModuLM will become a powerful tool for advancing molecular relational learning and interdisciplinary research, providing flexible and efficient technical support for future scientific endeavors.

6 Acknowledgement

This paper is partially supported by the National Natural Science Foundation of China (No.12227901). The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences., Anhui Science Foundation for Distinguished Young Scholars (No.1908085J24), Natural Science Foundation of China (No.62502491).

References

- [1] Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [2] Peizhen Bai, Filip Miljković, Bino John, and Haiping Lu. Interpretable bilinear attention network with domain adaptation improves drug–target prediction. *Nature Machine Intelligence*, 5(2):126–136, Feb 2023.
- [3] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.
- [4] Tianlang Chen, Shengjie Luo, Di He, Shuxin Zheng, Tie-Yan Liu, and Liwei Wang. Geom-former: A general architecture for geometric molecular representation learning. *arXiv preprint arXiv:2406.16853*, 2024.
- [5] Zhuo Chen, Jiahui Zhang, Sihan Wang, Hongxin Xiang, Jianmin Wang, Wenjie Du, and Yang Wang. Do-colm: Dynamic 3d conformation relationships capture with self-adaptive ordering molecular relational modeling in language models.
- [6] Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. Unifying molecular and textual representations via multi-task language modelling. *arXiv preprint arXiv:2301.12586*, 2023.
- [7] Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. Unifying molecular and textual representations via multi-task language modelling. In *International Conference on Machine Learning*, pages 6140–6157. PMLR, 2023.
- [8] Yunsie Chung, Florence H Vermeire, Haoyang Wu, Pierre J Walker, Michael H Abraham, and William H Green. Group contribution and machine learning approaches to predict abraham solute parameters, solvation free energy, and solvation enthalpy. *Journal of Chemical Information and Modeling*, 62(3):433–446, 2022.
- [9] Weitao Du, He Zhang, Yuanqi Du, Qi Meng, Wei Chen, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Se (3) equivariant graph neural networks with complete local frames. In *International Conference on Machine Learning*, pages 5583–5608. PMLR, 2022.
- [10] Wenjie Du, Shuai Zhang, Zhaohui Cai, Zhiyuan Liu, Junfeng Fang, Jianmin Wang, and Yang Wang. Molecular merged hypergraph neural network for explainable solvation free energy prediction. *Research*, 8:0740, 2025.
- [11] Wenjie Du, Shuai Zhang, Jun Xia Di Wu, Ziyuan Zhao, Junfeng Fang, and Yang Wang. 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*, pages 5808–5816, 2024.

[12] Yuanqi Du, Limei Wang, Dieqiao Feng, Guifeng Wang, Shuiwang Ji, Carla P Gomes, Zhi-Ming Ma, et al. A new perspective on building efficient and expressive 3d equivariant graph neural networks. *Advances in neural information processing systems*, 36:66647–66674, 2023.

[13] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015.

[14] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

[15] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

[16] Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, 2021.

[17] Junfeng Fang, Shuai Zhang, Chang Wu, Zhengyi Yang, Zhiyuan Liu, Sihang Li, Kun Wang, Wenjie Du, and Xiang Wang. Moltc: Towards molecular relational modeling in language models. *arXiv preprint arXiv:2402.03781*, 2024.

[18] Tianfan Fu, Cao Xiao, and Jimeng Sun. Core: Automatic molecule optimization using copy & refine strategy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 638–645, 2020.

[19] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

[20] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

[21] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[22] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[23] Laura M Grubbs, Mariam Saifullah, E Nohelli, Shulin Ye, Sai S Achi, William E Acree Jr, and Michael H Abraham. Mathematical correlations for describing solute transfer into functionalized alkane solvents containing hydroxyl, ether, ester or ketone solvents. *Fluid phase equilibria*, 298(1):48–53, 2010.

[24] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[25] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[26] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Advances in Neural Information Processing Systems*, 2 2021.

[27] Kexin Huang, Tianfan Fu, Lucas M Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. Deeppurpose: A deep learning library for drug–target interaction prediction. *Bioinformatics*, 36(22–23):5545–5547, Dec 2020.

[28] Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. Moltrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 2020.

[29] Kanchan Jha, Sriparna Saha, and Hiteshi Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):8360, 2022.

[30] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021.

[31] Joonyoung F Joung, Minhi Han, Minseok Jeong, and Sungnam Park. Experimental database of optical properties of organic compounds. *Scientific data*, 7(1):295, 2020.

[32] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. Pubchem 2023 update. *Nucleic Acids Res.*, 51(D1):1373–1380, 2023.

[33] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[34] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[35] Mario Krenn, Qianxiang Ai, Senja Barthel, Nessa Carson, Angelo Frei, Nathan C Frey, Pascal Friederich, Théophile Gaudin, Alberto Alexander Gayle, Kevin Maik Jablonka, et al. Selfies and the future of molecular string representations. *Patterns*, 3(10), 2022.

[36] Namkyeong Lee, Dongmin Hyun, Gyoung S. Na, Sungwon Kim, Junseok Lee, and Chanyoung Park. Conditional graph information bottleneck for molecular relational learning. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 18852–18871. PMLR, 2023.

[37] Namkyeong Lee, Dongmin Hyun, Gyoung S Na, Sungwon Kim, Junseok Lee, and Chanyoung Park. Conditional graph information bottleneck for molecular relational learning. *arXiv preprint arXiv:2305.01520*, 2023.

[38] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[39] Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. 3d-molm: Towards 3d molecule-text interpretation in language models. In *ICLR*, 2024.

[40] Youwei Liang, Ruiyi Zhang, Li Zhang, and Pengtao Xie. Drugchat: Towards enabling chatgpt-like capabilities on drug molecule graphs. *ArXiv*, abs/2309.03907, 2023.

[41] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

[42] Xuan Lin, Zhe Quan, Zhi-Jie Wang, Tengfei Ma, and Xiangxiang Zeng. Kggnn: Knowledge graph neural network for drug-drug interaction prediction. In *IJCAI*, volume 380, pages 2739–2745, 2020.

[43] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[44] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations (ICLR)*, 2022.

[45] Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*, 2023.

[46] Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International conference on learning representations (ICLR)*, 2022.

[47] Qi Mao, Hanxiao Wang, Roger W. H. Lau, Peter K. K. Loh, and Xiaorong Li. Gated multimodal units for information fusion. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 2314–2320, 2017.

[48] Aleksandr V Marenich, Casey P Kelly, Jason D Thompson, Gregory D Hawkins, Candee C Chambers, David J Giesen, Paul Winget, Christopher J Cramer, and Donald G Truhlar. Minnesota solvation database (mnsol) version 2012. 2020.

[49] David L Mobley and J Peter Guthrie. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design*, 28:711–720, 2014.

[50] Edouard Moine, Romain Privat, Baptiste Sirjean, and Jean-Noël Jaubert. Estimation of solvation quantities from experimental thermodynamic data: Development of the comprehensive compsol databank for pure and mixed solutes. *Journal of Physical and Chemical Reference Data*, 46(3), 2017.

[51] Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, Oct 2020.

[52] Gilchan Park, Sean McCorkle, Carlos Soto, Ian Blaby, and Shinjae Yoo. Extracting protein–protein interactions (ppis) from biomedical literature using attention-based relational context information. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 2052–2061. IEEE, 2022.

[53] Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. *arXiv preprint arXiv:2310.07276*, 2023.

[54] Ying Qian, Xinyi Li, Jian Wu, and Qian Zhang. Mcl-dti: Using drug multimodal information and bi-directional cross-attention learning method for predicting drug–target interaction. *BMC Bioinformatics*, 24(1), Aug 2023.

[55] Dan M Roden, Robert A Harrington, Athena Poppas, and Andrea M Russo. Considerations for drug interactions on qtc in exploratory covid-19 treatment. *Circulation*, 141(24):e906–e907, 2020.

[56] Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. Deep learning improves prediction of drug–drug and drug–food interactions. *Proceedings of the national academy of sciences*, 115(18):E4304–E4311, 2018.

[57] Tatsuya Sagawa and Ryosuke Kojima. Reactiont5: a large-scale pre-trained model towards application of limited reaction data. *arXiv preprint arXiv:2311.06708*, 2023.

[58] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[59] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021.

[60] Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in Neural Information Processing Systems*, 30:992–1002, 2017.

[61] Sizhe Sizhe Liu, Jun Xia, Lecheng Zhang, Yuchen Liu, Yue Liu, Wenjie Du, Zhangyang Gao, Bozhen Hu, Cheng Tan, Stan Z Li, et al. Flexmol: A flexible toolkit for benchmarking molecular relational learning. *Advances in Neural Information Processing Systems*, 37:35454–35467, 2024.

[62] Xiangru Tang, Andrew Tran, Jeffrey Tan, and Mark B. Gerstein. Mollm: A unified language model for integrating biomedical text with 2d and 3d molecular representations. *bioRxiv*, 2024.

[63] Nicholas P Tatonetti, Patrick P Ye, Roxana Daneshjou, and Russ B Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4(125):125ra31–125ra31, 2012.

[64] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

[65] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[66] Jithin John Varghese and Samir H Mushrif. Origins of complex solvent effects on chemical reactivity and computational tools to investigate them: a review. *Reaction Chemistry & Engineering*, 4(2):165–206, 2019.

[67] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[68] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[69] Florence H Vermeire and William H Green. Transfer learning for solvation free energies: From quantum chemistry to experiments. *Chemical Engineering Journal*, 418:129307, 2021.

[70] Taras Voitsitskyi, Roman Stratiichuk, Ihor Koleiev, Leonid Popryho, Zakhar Ostrovsky, Pavlo Henitsoi, Ivan Khropachov, Volodymyr Vozniak, Roman Zhytar, Diana Neschepurenko, and et al. 3dprotlda: A deep learning model for drug-target affinity prediction based on residue-level protein graphs. *RSC Advances*, 13(15):10261–10272, 2023.

[71] Sihan Wang, Shuai Zhang, Zhuo Chen, Xuqiang Li, Jianmin Wang, Wenjie Du, and Yang Wang. Trgh-ppi: Effective and generalized prediction of protein-protein interactions through transformer and graph. *IEEE Transactions on Computational Biology and Bioinformatics*, 2025.

[72] Fang Wu, Shuting Jin, Yinghui Jiang, Xurui Jin, Bowen Tang, Zhangming Niu, Xiangrong Liu, Qiang Zhang, Xiangxiang Zeng, and Stan Z Li. Pre-training of equivariant graph matching networks with conformation flexibility for drug binding. *Advanced Science*, 9(33):2203796, 2022.

[73] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, Aug 2019.

[74] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[75] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

[76] Mehdi Yazdani-Jahromi, Niloofar Yousefi, Aida Tayebi, Elayaraja Kolanthai, Craig J Neal, Sudipta Seal, and Ozlem Ozmen Garibay. Attentionsitedti: An interpretable graph-based model for drug-target interaction prediction using nlp sentence-level relation classification. *Briefings in Bioinformatics*, 23(4), Jul 2022.

[77] Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(862), 2022.

[78] D. D. Zhang, S. Xia, and Y. K. Zhang. Accurate prediction of aqueous free solvation energies using 3d atomic feature-based graph neural network with transfer learning. *Journal of Chemical Information and Modeling*, 62(8):1840–1848, 2022.

[79] Wen Zhang, Yanlin Chen, Feng Liu, Fei Luo, Gang Tian, and Xiaohong Li. Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data. *BMC bioinformatics*, 18:1–12, 2017.

[80] Xuan Zhang, Cheng Chen, Zhaoxu Meng, Zhenghe Yang, Haitao Jiang, and Xuefeng Cui. Coatgin: Marrying convolution and attention for graph-based molecule property prediction. In *2022 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 374–379. IEEE, 2022.

[81] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023.

[82] Yi Zhong, Xueyu Chen, Yu Zhao, Xiaoming Chen, Tingfang Gao, and Zuquan Weng. Graph-augmented convolutional networks on drug-drug interactions prediction. *arXiv preprint arXiv:1912.03702*, 2019.

[83] Yujie Zhong, Guangming Li, Jianxin Yang, et al. Learning motif-based graphs for drug–drug interaction prediction via local–global self-attention. *Nature Machine Intelligence*, 6:1094–1105, 2024.

[84] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: a universal 3d molecular representation learning framework. 2023.

[85] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. 2023.

[86] Lu Zhou and Rong-Hua Zhang. A self-attention–based neural network for three-dimensional multivariate modeling and its skillful enso predictions. *Science Advances*, 9(10):eadf2827, 2023.

[87] Yan Zhu, Lingling Zhao, Naifeng Wen, Junjie Wang, and Chunyu Wang. Datadta: A multi-feature and dual-interaction aggregation framework for drug–target binding affinity prediction. *Bioinformatics*, 39(9), Sep 2023.

[88] M Zitnik, R Sosi, S Maheshwari, and J Leskovec. Stanford biomedical network dataset collection. *Biosn. Datasets Stanford Biomed. Netw. Dataset Collect*, 2018.

A Appendix

A.1 General Information

A.1.1 Links

The code for ModuLM is currently available in our code repository <https://github.com/ssjhw/ModuLM>. The repository provides reproduction scripts for the best-performing model, and if you wish to reproduce other experiments conducted in this paper, please refer to the Appendix A.4.

A.1.2 Licenses

The ModuLM is under the CC BY 4.0. We, the authors, bear all responsibility in case of violation of rights.

A.2 Details of Datasets

In ModuLM, we provide diverse datasets for DDI, SSI, and CSI tasks to comprehensively evaluate ModuLM’s flexibility and generalizability. An overview of the datasets is provided in Table 8.

Table 8: Data statistics.

Task	Dataset	\mathcal{G}^1	\mathcal{G}^2	Pairs
DDI	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

ZhangDDI [79]. This dataset consists of 548 drugs with 48,548 labeled drug–drug interactions (DDIs). It integrates multiple drug similarity measures, such as chemical structure and target-based similarities, to enable similarity-aware DDI prediction. Due to its comprehensive annotation and focus on similarity information, ZhangDDI is widely used as a benchmark for evaluating similarity-based computational methods in drug interaction prediction.

ChChMiner [88]. ChChMiner provides a curated collection of 1,322 drugs and 48,514 interactions derived from FDA-approved labels and peer-reviewed literature. Emphasizing clinically validated DDIs, it serves as a reliable dataset for real-world risk assessment and pharmacovigilance studies. Its focus on verified clinical data makes it particularly valuable for safety-critical applications.

DeepDDI [56]. Comprising 192,284 labeled DDIs annotated with side effect profiles, DeepDDI leverages data from DrugBank to support multi-label classification of adverse drug interactions. This dataset enables models to predict not only the presence of interactions but also the specific adverse effects, thereby providing richer supervision for deep learning models in pharmacology.

TWOSIDES [63]. TWOSIDES features 555 drugs with over 3.5 million reported interactions encompassing 1,318 distinct interaction types. Extracted from FDA adverse event reporting systems, it offers a large-scale resource for investigating long-tail and rare DDI patterns. This dataset is instrumental for exploring subtle and complex interaction effects in post-market drug safety monitoring.

Chromophore [31]. This dataset includes 20,236 chromophore–solvent pairs with experimentally measured optical properties such as absorption wavelength, emission wavelength, and excited-state lifetime. Unreliable or outlier data entries have been rigorously filtered, and log-normalization is

applied to reduce skewness in target distributions. Chromophore enables in-depth studies of solvent effects on photophysical behavior at the molecular level.

MNSol [48]. MNSol contains 3,037 records of solvation or transfer free energies covering 790 solutes and 92 solvents. The dataset has been filtered to 2,275 high-quality solute–solvent combinations consistent with established benchmarks, providing a robust resource for training and evaluating solvation energy prediction models in computational chemistry.

FreeSolv [49]. FreeSolv offers 643 hydration free energy measurements of small molecules in aqueous solution, of which 560 experimental entries are retained after quality control. It serves as a standard benchmark for predicting aqueous solvation properties, critical for understanding solute–solvent interactions relevant to drug design and molecular biology.

CompSol [50]. CompSol focuses on 3,548 solute–solvent pairs emphasizing the role of hydrogen bonding in solvation energy. By targeting molecular interactions specifically related to hydrogen bonds, this dataset provides a fine-grained benchmark to evaluate models’ ability to capture solvent effects driven by directional intermolecular forces.

Abraham [23]. Based on Abraham’s linear solvation energy relationships, this dataset compiles 6,091 solute–solvent combinations from 1,038 solutes and 122 solvents. It supports solvent effect modeling through well-established physicochemical parameters, making it suitable for studies on quantitative structure–property relationships (QSPR) and solvent screening.

CombiSolv [69]. CombiSolv integrates four benchmark datasets — MNSol, FreeSolv, CompSol, and Abraham — into a comprehensive collection of 10,145 solute–solvent pairs. This unified dataset facilitates broader generalization and benchmarking of molecular solvation models across diverse chemical spaces and experimental conditions.

CombiSolv-QM [69]. CombiSolv-QM extends CombiSolv by including 1 million solute–solvent pairs generated via quantum mechanical (QM) simulations. This large-scale synthetic dataset enables the evaluation of model robustness and scalability, providing a valuable resource for training models that generalize well to chemically diverse and complex molecular systems.

A.3 Details of ModuLM Components

A.3.1 2D Graph Encoders

GCN [34]. Graph Convolutional Network introduces a spectral-based graph convolution operation by approximating localized filters with a normalized graph Laplacian. This enables efficient aggregation of neighboring node features, allowing semi-supervised learning directly on graph-structured data. GCN’s simple yet powerful layer design laid foundational groundwork for many subsequent graph neural network models by balancing computational efficiency and expressiveness.

MPNN [22]. Message Passing Neural Network generalizes the idea of graph convolutions by explicitly learning both the message functions exchanged between nodes and the update functions that revise node states. This flexible framework supports incorporation of domain-specific knowledge, allowing improved molecular property predictions by capturing complex interactions through iterative message passing.

GAT [68]. Graph Attention Network innovatively applies self-attention mechanisms on graph neighborhoods, enabling the model to assign adaptive importance weights to different neighbors during embedding updates. This attention mechanism enhances the capacity to learn from irregular and heterogeneous graph structures, improving representation quality without requiring explicit graph preprocessing.

NeuralFP [13]. Neural Fingerprints pioneer end-to-end learning of molecular fingerprints using graph convolutional layers. Unlike handcrafted descriptors, this method automatically extracts relevant chemical features for property prediction tasks, showing superior performance and adaptability across diverse molecular datasets.

AttentiveFP [73]. Attentive Fingerprint leverages attention mechanisms at both atom and substructure levels, allowing the model to focus selectively on chemically important regions. This hierarchical attention not only improves prediction accuracy but also enhances interpretability, making it valuable for QSAR modeling and drug discovery.

GIN [74]. Graph Isomorphism Network achieves maximum discriminative power among GNNs by using injective neighborhood aggregation functions. This theoretical guarantee enables GIN to distinguish graph structures as effectively as the Weisfeiler-Lehman graph isomorphism test, pushing forward the limits of GNN expressiveness.

GraphSAGE [25]. Graph Sample and AggregatE introduces an inductive learning framework that samples and aggregates neighborhood features to generate node embeddings. This scalable approach enables GNNs to handle large, evolving graphs efficiently, supporting applications where graphs are dynamic or partially observed.

CoATGIN [80]. CoATGIN innovatively combines convolutional aggregation with attention mechanisms, capturing both local subgraph motifs and global graph contexts in molecular graphs. This hybrid approach enhances molecular representation quality by balancing locality and global awareness, beneficial for tasks requiring nuanced structural understanding.

A.3.2 3D Conformation Encoders

EGNN [58]. Equivariant Graph Neural Network maintains equivariance to Euclidean transformations by jointly updating node features and their spatial coordinates. This design allows effective modeling of 3D molecular geometry without sacrificing rotational or translational invariance, crucial for accurate physical property predictions.

3D-GeoFormer [86]. 3D-GeoFormer employs geometric transformers to model spatial relationships in molecular structures, leveraging attention mechanisms adapted for 3D spatial data. This enables capturing long-range dependencies and complex geometric interactions, advancing 3D molecular representation learning beyond local neighborhoods.

SE(3)-Transformer [19]. SE(3)-Transformer integrates tensor field networks to ensure full SE(3) equivariance, allowing seamless handling of rotations and translations in 3D space. This makes it particularly suitable for modeling complex molecular and protein structures where geometric consistency under transformations is critical.

PaiNN [59]. PaiNN achieves rotational equivariance by explicitly separating scalar and vector features within its message-passing framework. This distinction enables precise modeling of directional physical quantities such as forces and energies, leading to state-of-the-art accuracy in quantum chemistry tasks.

GVP [30]. Geometric Vector Perceptrons combine scalar and vector features through specialized geometric transformations, enabling effective encoding of molecular structures in 3D. GVP generalizes across various 3D molecular graphs and shows strong performance on both structural and functional prediction tasks.

GearNet [81]. GearNet builds hierarchical molecular representations by integrating multi-scale structural and sequential information, capturing both local atomic details and broader biological context. This approach enhances model capability across different biological levels, supporting tasks like protein structure prediction and interaction analysis.

DimeNet++ [21]. DimeNet++ improves directional message passing by explicitly encoding angular and directional information between atoms. Its architecture achieves higher accuracy and efficiency compared to its predecessor, effectively modeling molecular interactions that depend on precise geometric orientations.

SchNet [60]. SchNet introduces continuous-filter convolutional layers to model quantum interactions in molecular systems. Its end-to-end differentiable framework enables accurate predictions of atomic-level properties by capturing spatial correlations without requiring predefined features.

SphereNet [44]. SphereNet encodes spherical coordinates such as angles and radial distances to better capture 3D molecular geometry. This precise geometric encoding leads to improved performance on quantum property prediction tasks by more faithfully representing spatial relations.

G-SphereNet [46]. G-SphereNet extends SphereNet by incorporating an autoregressive generative process for molecular conformations, enabling both accurate representation and generation of complex 3D structures. This combination supports downstream applications in molecular design and simulation.

Uni-Mol [85]. Uni-Mol unifies pretraining and finetuning in 3D molecular representation learning via a transformer-based architecture. Its spatially aware design supports a wide variety of downstream tasks, improving generalization by leveraging large-scale molecular conformations during pretraining.

GeoMFormer [4]. GeoMFormer proposes a general transformer framework tailored to capture geometric relationships within molecular structures. By effectively modeling 3D spatial interactions, it improves the expressiveness and transferability of molecular representations across diverse tasks.

GotenNet [1]. GotenNet rethinks geometric message passing by designing an efficient 3D equivariant graph neural network that significantly reduces computational cost while maintaining or improving predictive accuracy. Its novel architectural choices enable scalable and effective modeling of complex molecular geometries.

A.3.3 Interaction Layers

Bilinear Attention [2]. The Bilinear Attention Network (BAN) layer captures interactions between 2D feature sets through bilinear transformations, followed by attention pooling and batch normalization.

Self Attention [67]. Self-attention mechanisms allow a feature set to focus on its own elements, enabling models to capture relationships within the same source of data.

Cross Attention [54]. Cross-attention captures interactions between two distinct feature sets by applying attention mechanisms that focus on cross-source dependencies.

Highway [87]. The Highway mechanism combines 1D features through gated layers, allowing information to flow selectively by controlling the gates.

Gated Fusion [47]. Gated fusion combines 1D features from two sources by applying gated transformations, producing a unified representation that captures the interactions between them.

Bilinear Fusion [41]. Bilinear Fusion combines 1D features using a bilinear transformation and ReLU activation, capturing multiplicative interactions to enhance feature representation.

Mean. The Mean method combines feature sets by averaging their values.

A.3.4 Backbones

In ModuLM, we provide seven popular LLMs: DeepSeek-1.5B, DeepSeek-7B, DeepSeek-14B [3], LLaMA-1B, LLaMA-13B [65], Galactica-1.3B, and Galactica-6.7B [64].

DeepSeek. The DeepSeek-1.5B, DeepSeek-7B, and DeepSeek-14B models are derived from the Qwen-2.5 series, which were originally licensed under the Apache 2.0 License, and have now been fine-tuned with 800k samples curated using DeepSeek-R1.

LLama. LLama 1B incorporated logits from the LLama 3.1 8B and 70B models during the pretraining stage, using the outputs (logits) from these larger models as token-level targets. Knowledge distillation was applied after pruning to recover performance. LLama-13B was pretrained on 2 trillion tokens of data from publicly available sources and fine-tuned on publicly available instruction datasets, along with over one million new human-annotated examples, making it a general-purpose LLM.

Galactica. Galactica-1.3B and Galactica-6.7B are large language models developed by Meta for scientific research and knowledge-intensive tasks. These models are designed to assist with tasks in fields like scientific literature, research summarization, and computational biology.

A.3.5 Prompts

In the main text, we mentioned the two prompts involved in our experiments: Direct and Chain-of-Thought-based reasoning. The prompt design is shown in the Table 9.

A.4 Example Usage of ModuLM

This section demonstrates how to construct example models from the main text using ModuLM. In ModuLM, we provide a configuration method based on a JSON file. In the modules we offer, users only need to modify the corresponding parameters. It is worth noting that some of the parameters

Table 9: Prompts settings.

Direct Reasoning for Qualitative Tasks
The first molecule <SMILES0> and the second molecule <SMILES1> are expected to interact with each other, potentially forming a molecular complex or influencing each other's properties.
CoT-based Reasoning for Qualitative Tasks
The first molecule <SMILES0> is likely to exhibit [Property0], while the second molecule <SMILES1> is likely to exhibit [Property1]. Hence, the first drug molecule may alter the therapeutic effects of the second drug molecule. Therefore, they are likely to interact with each other.
Direct Reasoning for Quantitative Tasks
The solvation Gibbs free energy between the first molecule <SMILES0> and the second molecule <SMILES1> is 4.6232.
CoT-based Reasoning for Quantitative Tasks
The first molecule <SMILES0> is likely to exhibit [Property0], while the second molecule <SMILES1> is likely to exhibit [Property1]. Therefore, their solvation Gibbs free energy is likely to fall between 4.0 and 4.5, with a precise value potentially being 4.6232.

below are included solely to demonstrate the comprehensiveness of our framework; if the goal is simply to use it, most hyperparameters do not need to be changed.

A.4.1 Loading the Dataset

Taking the DeepDDI dataset loading as an example, the path to the dataset is defined using the `root` parameter. To accelerate data loading, the `num_workers` parameter is used to enable multi-threaded data processing. Additionally, the `use_3d` flag controls whether to incorporate 3D molecular conformational data as input. This allows users to flexibly switch between 2D and 3D molecular representations depending on the task requirements and available structural information.

```
{
  "root": "data/DDI/DeepDDI/",
  "num_workers": 5,
  "use_3d": true
}
```

A.4.2 Initializing Encoder

In this setup, we utilize the Uni-Mol model as our 3D molecular conformation encoder. The specific encoder is selected by setting the `graph3d` parameter accordingly. To fine-tune its behavior and architecture, we provide a dedicated configuration file that defines key hyperparameters such as the number of layers, embedding dimensions, attention mechanisms, and dropout rates. This modular design allows for flexible customization and seamless integration into various molecular representation learning tasks.

```
{
  "graph3d": "unimol",
  "con_activation_dropout": 0.0,
  "con_activation_fn": "gelu",
  "con_attention_dropout": 0.1,
  "con_delta_pair_repr_norm_loss": -1.0,
  "con_dropout": 0.1,
  "con_emb_dropout": 0.1,
  "con_encoder_attention_heads": 64,
  "con_encoder_embed_dim": 512,
  "con_encoder_ffn_embed_dim": 2048,
  "con_encoder_layers": 15,
  "con_max_atoms": 256,
  "con_max_seq_len": 512
}
```

It is important to note that in order to build the example model provided in the main text, we need to make further modifications and extensions to the Uni-Mol model. Once the code has been extended, we can simply place it in the specified directory and continue managing and calling it through the config file.

A.4.3 Configuring LLM

Here, the `mode` specifies the model's mode, whether it is pretraining or fine-tuning. The `backbone` parameter indicates the LLM to be used, while `min_len` and `max_len` define the minimum and maximum lengths of the generated text. Additional details and parameters for LLM-based text generation are provided in our code repository; this section highlights only a few key settings as examples.

```
{
  "mode": "ft",
  "backbone": "DeepSeek-1.5B",
  "min_len": 10,
  "max_len": 40
}
```

A.4.4 Training the Model

After constructing the model, we can fine-tune it by configuring the appropriate LoRA file. To make it easier for others to fine-tune using our framework, we provide the LoRA parameter configuration for model training here.

```
{
  "base_model_name_or_path": null,
  "bias": "none",
  "fan_in_fan_out": false,
  "inference_mode": false,
  "init_lora_weights": true,
  "lora_alpha": 32,
  "lora_dropout": 0.1,
  "target_modules": ["q_proj", "v_proj", "out_proj", "fc1", "fc2"],
  "peft_type": "LORA",
  "r": 16,
  "modules_to_save": null,
  "task_type": "CAUSAL_LM"
}
```

The specific batch size and number of training epochs are also configured in a unified manner. Here, the `batch_size` specifies the number of samples processed in each training step, which in this case is set to 12. The `max_epochs` defines the total number of training iterations over the entire dataset, set here to 20 epochs. The `save_every_n_epochs` parameter indicates that the model's state will be saved every 5 epochs. The `scheduler` field specifies the learning rate scheduling strategy—`linear_warmup_cosine_lr` gradually increases the learning rate during a warm-up period, then decays it following a cosine curve. The `seed` ensures reproducibility of training results by fixing randomness. `warmup_lr` and `warmup_steps` define the initial learning rate and the number of steps over which it will warm up, respectively. Lastly, `weight_decay` is used as a regularization technique to prevent overfitting by penalizing large weights during optimization.

```
{
  "batch_size": 12,
  "max_epochs": "30",
  "save_every_n_epochs": 5,
  "scheduler": "linear_warmup_cosine_lr",
  "seed": 42,
  "warmup_lr": 1e-06,
  "warmup_steps": 1000,
  "weight_decay": "0.05"
}
```

A.5 More experimental details

A.5.1 Details of Experimental Setup

In this section, we provide a more detailed description of the experimental data and testing configurations used in the main text.

Training Epochs. At the beginning of each experiment, we initiate incremental pretraining by running 5 epochs on the collected pretraining dataset. During the subsequent fine-tuning stage, the number of training epochs is task-dependent. Specifically, for the DDI task, we fine-tune the model for 50 epochs. For SSI datasets containing more than 3000 molecular pairs, we adopt a two-stage fine-tuning strategy: first, the model is fine-tuned on the CombiSolv-QM dataset for 100 epochs, followed by an additional 30 epochs on the target dataset. In contrast, for SSI datasets with fewer than 3000 molecular pairs, the fine-tuning stage is shortened to 20 epochs. Notably, both pretraining and fine-tuning phases share the same optimizer and learning rate scheduling configurations, as described in the following section. It is worth noting that during training, we did not employ an early stopping strategy. Instead, we saved the model’s performance on both the validation and test sets after each epoch to facilitate more thorough analysis. When reporting the results, we selected the test set performance corresponding to the epoch with the best validation set performance for evaluation.

Training Strategy. We employ the AdamW optimizer with a weight decay coefficient of 0.05 to mitigate overfitting and stabilize training. The learning rate is governed by a linear warm-up followed by cosine decay schedule, which helps accelerate convergence in the early stages and enables refined optimization during the later phases. To further enhance efficiency and reduce training overhead, we adopt Low-Rank Adaptation (LoRA), implemented using the OpenDelta and PEFT libraries. The rank parameter of LoRA is set to $r = 16$. For models in the DeepSeek series, LoRA is applied to the following modules: [q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj]. For the LLaMA and Galactica models, LoRA is instead integrated into [q_proj, v_proj, out_proj, fc1, fc2].

A.5.2 The Impact of Post Pretraining

Table 10: Performance of different Post Pretraining Stragedy.

Experiment	Accuracy (ChChMiner)	Accuracy (ZhangDDI)	Accuracy (DeepDDI)	RMSE (FreeSolv)	RMSE (CompSol)	RMSE (CombiSolv)
Molecular Interaction-based	0.962 ± 0.007	0.911 ± 0.006	0.954 ± 0.008	0.682 ± 0.017	0.301 ± 0.016	0.374 ± 0.019
Substructure-based	0.961 ± 0.007	0.908 ± 0.006	0.956 ± 0.007	0.692 ± 0.017	0.295 ± 0.017	0.372 ± 0.019
Structure Similarity-guided	0.966 ± 0.004	0.914 ± 0.005	0.961 ± 0.008	0.676 ± 0.020	0.292 ± 0.018	0.367 ± 0.022

In the main text, we integrated three incremental pretraining methods to help the model better acquire domain knowledge relevant to molecular relational learning. To evaluate the impact of these pretraining strategies on model performance, we conducted a series of tests. Specifically, we selected the best-performing model from the main text as the evaluation baseline. Since conducting zero-shot evaluations using pure textual input can lead to unreliable or unreportable results on certain datasets, we opted to retain the fine-tuning process and only replace the pretraining strategy during testing. It is important to note that our experiments in the main text have already demonstrated that DeepSeek consistently outperforms other models under the same pretraining settings. Therefore, we conduct all evaluations here using the DeepSeek-1.5B model. The results are shown in Table 10.

A.5.3 Specialized Encoder Analysis

To better compare the performance differences introduced by various encoders, we standardize all other configurations and only replace the encoder components for analysis. The experimental setup follows a similar design to that used in the main body, employing an MLP for modality alignment and using DeepSeek-1.5B as the backbone LLM. It is important to emphasize that the main contribution of this work is to provide a fair, flexible, and extensible benchmark framework for MRL. Given the combinatorial complexity of possible encoder and architecture choices, our focus is on evaluation rather than theoretical analysis. We believe that performance differences stemming from specific encoders or architectures should be thoroughly analyzed and justified by the authors of the respective methods. As mentioned in the main text, 3D molecular conformations contain richer structural information; therefore, to aid readers in gaining a deeper understanding of the MRL task, we provide

a brief comparative discussion of Uni-Mol and GotenNet as representative examples to highlight structural design differences and their potential impact on performance. Overall performance is shown in Table 11.

Table 11: Overall Performance of different Encoders.

Experiment	Accuracy (ChChMiner)	Accuracy (ZhangDDI)	Accuracy (DeepDDI)	RMSE (FreeSolv)	RMSE (CompSol)	RMSE (CombiSolv)
Uni-Mol	0.959 ± 0.010	0.903 ± 0.004	0.952 ± 0.007	0.690 ± 0.021	0.310 ± 0.019	0.387 ± 0.023
GotenNet	0.966 ± 0.004	0.914 ± 0.005	0.961 ± 0.008	0.676 ± 0.020	0.292 ± 0.018	0.367 ± 0.022

GotenNet outperforms Uni-Mol primarily due to its innovative design in modeling three-dimensional geometric information. Firstly, GotenNet employs an efficient and refined geometric message-passing mechanism that captures spatial relationships and angular dependencies within molecules more accurately, enabling a more comprehensive understanding of molecular structural complexity. Secondly, while maintaining SE(3) equivariance, GotenNet optimizes computational efficiency by reducing redundant calculations, thereby improving both training and inference speed without compromising representational capacity. Moreover, GotenNet places greater emphasis on multi-scale feature integration, combining local details with global conformational information, which enhances its generalization ability across various molecular property prediction tasks. In contrast, although Uni-Mol also utilizes a transformer-based 3D encoding strategy, its handling of geometric information is comparatively coarser and incurs higher computational costs, limiting the model’s scalability and performance improvements. Therefore, through structural innovations and efficiency optimizations, GotenNet achieves a deeper understanding of 3D molecular information, leading to superior performance. The detailed computational efficiency can be found in Table 12. Here, we use the average number of samples processed per second on a single GPU as the evaluation metric.

Table 12: Efficiency of different encoders.

Model	Uni-Mol	GotenNet
Rate	2.74 it/s	2.66 it/s

Table 13: Performance of different encoders on molecules of varying sizes.

Experiment	344	566	628	730	1846
Uni-Mol	0.879 ± 0.012	0.863 ± 0.014	0.852 ± 0.019	0.833 ± 0.020	0.739 ± 0.022
GotenNet	0.891 ± 0.010	0.884 ± 0.013	0.873 ± 0.018	0.845 ± 0.019	0.752 ± 0.021

To further validate whether the performance differences between GotenNet and Uni-Mol across molecules of varying sizes align with their respective architectural designs, we conducted additional experiments. We combined ZhangDDI, ChChDDI, DeepDDI, and TWO SIDES into a large aggregated dataset, grouping the samples by molecular mass with 20,000 samples per group. Here, AM denotes the average molecular mass within each group. Accuracy was used as the evaluation metric. For detailed performance results, please refer to the Table 13.

A.5.4 Computational Efficiency Analysis

To better compare the differences in computational efficiency brought by various backbones, encoding methods, and encoders, we conducted a more systematic time efficiency comparison following a similar approach as described above. The detailed results are shown in Table 15. To better highlight the differences, we selectively tested a subset of configurations presented in the main text. The details are shown in Table 14.

From the Table 15, we can observe that the primary factor affecting computational efficiency across different model configurations is the choice of backbone. Larger backbones significantly increase the computational cost. The second key factor is modality fusion—integrating more modalities increases processing complexity, thereby reducing efficiency. Notably, when using Q-former for alignment, the performance overhead becomes especially pronounced due to the large number of trainable parameters typically involved. For example, in configuration 1.16*, where all modalities are fused and Q-former is used for alignment, the processing rate drops to a very low level. Therefore, this

Table 14: Experimental Settings on DDI Datasets

Experiment No.	Backbone	Encoder	Interaction	Input Feature
1.2	Galactica-1.3B	GIN	-	$m_s + m_g$
1.3	Galactica-1.3B	GIN	Cross Attention	$m_s + m_g$
1.6	DeepSeek-1.5B	-	-	m_s
1.7	DeepSeek-1.5B	GIN	-	$m_s + m_g$
1.8	DeepSeek-1.5B	GotenNet	-	$m_s + m_c$
1.10	DeepSeek-14B	GotenNet	-	$m_s + m_c$
1.16	DeepSeek-1.5B	GIN+GotenNet	-	$m_s + m_g + m_c$

Note: m_s = molecular sequence, m_g = molecular graph, m_c = molecular conformation. '-' indicates that no method is applied.

Table 15: Efficiency of different encoders. * indicates the use of Q-former for alignment.

Model	1.2	1.3	1.6	1.7	1.7*	1.8	1.8*	1.10	1.16	1.16*
Rate	3.16 it/s	3.10 it/s	4.22 it/s	3.38 it/s	2.98 it/s	2.66 it/s	2.28 it/s	1.22 it/s	2.01 it/s	1.11 it/s

table is intended to help users of our framework make informed decisions about model configurations, enabling a balanced trade-off between computational efficiency and model performance.

A.5.5 Impact of Molecular Dataset

Meanwhile, to facilitate the exploration of model adaptability across different dataset scales and molecular sizes under various configurations, we first conducted experiments by grouping molecules based on their sizes, following a similar approach as described above. Additionally, we performed experiments using sampled subsets of varying scales from the ZhangDDI dataset for comparative analysis. Detailed experimental results can be found in Tables 16 and 17.

Table 16: Model performance variation with respect to molecular size, using Accuracy as the evaluation metric.

Model	1.2	1.3	1.6	1.7	1.7*	1.8	1.8*	1.10	1.16
334	0.868 _(.014)	0.874 _(.012)	0.855 _(.019)	0.879 _(.017)	0.881 _(.015)	0.892 _(.014)	0.890 _(.015)	0.860 _(.013)	0.851 _(.017)
566	0.851 _(.015)	0.863 _(.014)	0.843 _(.018)	0.864 _(.016)	0.869 _(.015)	0.872 _(.016)	0.865 _(.017)	0.845 _(.014)	0.837 _(.016)
628	0.825 _(.016)	0.842 _(.015)	0.818 _(.017)	0.838 _(.018)	0.840 _(.016)	0.845 _(.015)	0.832 _(.016)	0.820 _(.015)	0.812 _(.017)
730	0.790 _(.018)	0.804 _(.017)	0.780 _(.019)	0.812 _(.017)	0.815 _(.014)	0.822 _(.013)	0.809 _(.014)	0.785 _(.016)	0.778 _(.018)
1846	0.701 _(.019)	0.714 _(.017)	0.687 _(.020)	0.721 _(.011)	0.717 _(.012)	0.754 _(.010)	0.747 _(.009)	0.690 _(.019)	0.742 _(.016)

The data presented in the table clearly indicate that, regardless of the model combinations employed, the capability to process large molecules falls short of expectations. This limitation is not unique to our approach; it is a common challenge observed across existing multimodal molecular representation models. The sheer size of large molecules not only leads to a significant decrease in computational efficiency but also introduces an overwhelming amount of complex information. Current encoding models struggle to effectively manage this complexity. In the context of LLMs, our methodology involves simply concatenating the molecular text modality with its corresponding multimodal data before feeding it into the LLM for inference. This approach aligns with the prevailing standard for multimodal LLMs. However, when the multimodal data cannot be encoded effectively, it may actually disrupt or confuse the LLM’s decision-making process. Additionally, the excessively long molecular expressions associated with large molecules themselves pose further challenges, negatively impacting the LLM’s ability to generate accurate predictions. Overall, these factors highlight fundamental bottlenecks in scaling current models to handle large molecular structures effectively.

Due to the limitations of multimodal models, we are unable to perform zero-shot testing on them, which would lead to a complete collapse in model performance. Based on the data above, we observe that when training with LLMs, an insufficient amount of data often prevents the model from learning meaningful representations. This issue becomes even more pronounced in multimodal LLMs. Without exposing the model to enough diverse molecular examples from different modalities during training, the LLM is unable to accurately understand or infer the underlying semantics of the encoded information. In our experiments, we found that when using a zero-shot setting—i.e.,

Table 17: Model performance variation with respect to datasets size, using Accuracy as the evaluation metric.

Size	1.2	1.3	1.6	1.7	1.7*	1.8	1.8*	1.10	1.16
1000	0.645 _(.012)	0.650 _(.013)	0.641 _(.012)	0.673 _(.011)	0.675 _(.015)	0.703 _(.012)	0.687 _(.015)	0.681 _(.014)	0.697 _(.014)
5000	0.698 _(.013)	0.704 _(.013)	0.687 _(.014)	0.721 _(.012)	0.726 _(.014)	0.745 _(.013)	0.735 _(.014)	0.728 _(.013)	0.742 _(.012)
10000	0.772 _(.014)	0.779 _(.012)	0.761 _(.013)	0.793 _(.012)	0.799 _(.014)	0.818 _(.013)	0.808 _(.013)	0.802 _(.012)	0.815 _(.013)
20000	0.796 _(.014)	0.802 _(.012)	0.786 _(.014)	0.817 _(.012)	0.821 _(.013)	0.842 _(.012)	0.835 _(.013)	0.828 _(.012)	0.840 _(.012)
50000	0.866 _(.017)	0.870 _(.014)	0.841 _(.009)	0.886 _(.017)	0.881 _(.015)	0.902 _(.013)	0.897 _(.017)	0.860 _(.013)	0.892 _(.012)

directly testing the model without any prior training on our specific data—the LLM occasionally produced incoherent or grammatically incorrect outputs, indicating a lack of grounding in the input structure. Such behavior is clearly undesirable. However, as shown in the table, when the training data size is controlled and kept roughly consistent across settings, the relative ranking of model configurations in terms of performance remains largely unchanged. This suggests that the key differences between models lie not only in how much data they are exposed to, but also in how effectively their encoders are able to communicate molecular information to the LLM. In other words, a model’s ability to maintain performance across varying data sizes reflects the encoder’s competence in shaping representations that are both interpretable and informative to the LLM.

A.5.6 More Experimental Results

Due to space limitations in the main text, we present additional experimental results here, with the experimental setup consistent with the one described in the main text.

Table 18: More Results of DDI Datasets

Experiment	AUC-ROC (Drugbank)	Accuracy (Drugbank)	AUC-ROC (TWOSIDES)	Accuracy (TWOSIDES)
Chem T5[7]	0.921 \pm 0.010	0.859 \pm 0.013	0.906 \pm 0.015	0.856 \pm 0.022
MolCA[45]	0.934 \pm 0.018	0.898 \pm 0.010	0.942 \pm 0.014	0.907 \pm 0.015
MolT5[15]	0.930 \pm 0.016	0.904 \pm 0.018	0.940 \pm 0.013	0.929 \pm 0.017
MolTC[17]	0.978 \pm 0.006	0.951 \pm 0.005	0.980 \pm 0.005	0.970 \pm 0.007
1.1	0.933 \pm 0.011	0.891 \pm 0.012	0.912 \pm 0.017	0.877 \pm 0.014
1.2	0.945 \pm 0.010	0.922 \pm 0.011	0.957 \pm 0.009	0.923 \pm 0.009
1.3	0.950 \pm 0.009	0.935 \pm 0.010	0.950 \pm 0.008	0.926 \pm 0.008
1.4	0.955 \pm 0.008	0.938 \pm 0.009	0.946 \pm 0.007	0.935 \pm 0.008
1.5	0.946 \pm 0.010	0.931 \pm 0.010	0.951 \pm 0.008	0.918 \pm 0.008
1.6	0.938 \pm 0.016	0.901 \pm 0.012	0.920 \pm 0.017	0.893 \pm 0.018
1.7	0.963 \pm 0.007	0.944 \pm 0.007	0.970 \pm 0.006	0.952 \pm 0.006
1.8	0.975 \pm 0.006	0.950 \pm 0.006	0.982 \pm 0.005	0.975 \pm 0.005
1.9	0.950 \pm 0.008	0.937 \pm 0.008	0.971 \pm 0.006	0.949 \pm 0.006
1.10	0.940 \pm 0.010	0.926 \pm 0.010	0.961 \pm 0.007	0.938 \pm 0.007
1.11	0.920 \pm 0.014	0.886 \pm 0.010	0.907 \pm 0.021	0.855 \pm 0.020
1.12	0.935 \pm 0.011	0.920 \pm 0.010	0.945 \pm 0.012	0.911 \pm 0.008
1.13	0.947 \pm 0.008	0.933 \pm 0.009	0.953 \pm 0.013	0.923 \pm 0.009
1.14	0.952 \pm 0.007	0.931 \pm 0.009	0.959 \pm 0.012	0.932 \pm 0.010
1.15	0.935 \pm 0.019	0.877 \pm 0.020	0.927 \pm 0.010	0.894 \pm 0.011
1.16	0.971 \pm 0.008	0.948 \pm 0.004	0.984 \pm 0.007	0.978 \pm 0.003
Custom Model	0.982 \pm 0.010	0.956 \pm 0.007	0.986 \pm 0.007	0.980 \pm 0.009

The results presented in Table 19 and Table 18 are largely consistent with those reported in the main text, reaffirming the trends observed across different configurations. Notably, the introduction of additional modality information consistently yields substantial improvements in model performance, highlighting the effectiveness of leveraging multimodal signals in enhancing representation learning and generalization. In contrast, architectural modifications that increase model complexity—such

Table 19: More Results of SSI Datasets

Experiment	MAE (MNSol)	RMSE (MNSol)	MAE (Abraham)	RMSE (Abraham)
Chem T5[7]	0.537 ± 0.092	1.011 ± 0.083	0.621 ± 0.027	0.918 ± 0.032
MolCA[45]	0.511 ± 0.034	0.956 ± 0.049	0.580 ± 0.026	0.910 ± 0.032
MolT5[15]	0.466 ± 0.067	0.867 ± 0.069	0.544 ± 0.028	0.833 ± 0.029
MolTC[17]	0.354 ± 0.018	0.625 ± 0.023	0.211 ± 0.018	0.390 ± 0.021
1.1	0.510 ± 0.051	0.971 ± 0.063	0.572 ± 0.024	0.865 ± 0.030
1.2	0.451 ± 0.042	0.890 ± 0.028	0.524 ± 0.022	0.813 ± 0.027
1.3	0.436 ± 0.018	0.877 ± 0.027	0.496 ± 0.024	0.804 ± 0.029
1.4	0.410 ± 0.017	0.808 ± 0.024	0.447 ± 0.020	0.681 ± 0.023
1.5	0.506 ± 0.036	0.944 ± 0.043	0.522 ± 0.030	0.787 ± 0.031
1.6	0.502 ± 0.045	0.936 ± 0.052	0.591 ± 0.028	0.863 ± 0.025
1.7	0.386 ± 0.034	0.727 ± 0.035	0.394 ± 0.026	0.512 ± 0.029
1.8	0.343 ± 0.017	0.618 ± 0.024	0.204 ± 0.017	0.408 ± 0.021
1.9	0.488 ± 0.029	0.834 ± 0.031	0.416 ± 0.027	0.562 ± 0.029
1.10	0.501 ± 0.030	0.851 ± 0.032	0.456 ± 0.018	0.618 ± 0.022
1.11	0.499 ± 0.016	0.859 ± 0.023	0.437 ± 0.021	0.620 ± 0.030
1.12	0.550 ± 0.047	1.118 ± 0.064	0.614 ± 0.029	0.933 ± 0.030
1.13	0.475 ± 0.036	0.878 ± 0.033	0.511 ± 0.019	0.764 ± 0.021
1.14	0.461 ± 0.021	0.818 ± 0.027	0.489 ± 0.020	0.713 ± 0.021
1.15	0.597 ± 0.031	0.864 ± 0.035	0.422 ± 0.021	0.683 ± 0.023
1.16	0.352 ± 0.020	0.625 ± 0.018	0.201 ± 0.015	0.389 ± 0.017
Custom Model	0.340 ± 0.011	0.601 ± 0.010	0.199 ± 0.008	0.379 ± 0.011

as altering internal modules or adding more parameters—do lead to moderate performance gains. However, these improvements are generally less pronounced compared to those achieved through the integration of new modalities. This suggests that the diversity and complementarity of multimodal data play a more critical role than mere architectural sophistication in driving performance gains.

In addition to the DDI and SSI datasets, we further evaluate our framework on the CSI dataset to demonstrate its comprehensiveness and scalability. It is worth noting that the performance of models under different configurations on the CSI dataset varies significantly. To facilitate a more intuitive comparison of the performance across different settings, we report the best-performing configuration for each backbone. The results are summarized in the Table 20. Note that the three datasets in the CSI domain are all derived by splitting the Chromophore dataset.

Table 20: Performance on CSI Datasets

Experiment	MAE (Absorption)	RMSE (Absorption)	MAE (Emission)	RMSE (Emission)	MAE (Lifetime)	RMSE (Lifetime)
MolTC[17]	17.55 ± 1.83	29.10 ± 2.15	20.22 ± 1.91	34.17 ± 2.02	0.911 ± 0.052	1.213 ± 0.092
1.4	18.67 ± 2.01	31.33 ± 2.47	22.37 ± 2.01	36.71 ± 2.93	1.011 ± 0.061	1.502 ± 0.103
1.8	16.71 ± 1.82	28.67 ± 1.99	19.08 ± 1.89	38.00 ± 1.87	0.943 ± 0.054	1.119 ± 0.077
1.14	19.01 ± 2.01	32.46 ± 2.92	21.84 ± 1.96	37.56 ± 3.02	1.009 ± 0.060	1.711 ± 0.095
Custom Model	15.42 ± 1.53	27.65 ± 1.91	17.11 ± 1.68	31.55 ± 1.60	0.929 ± 0.064	1.123 ± 0.082

From the data in the Table above, we can clearly analyze the performance differences of various models under different experimental settings. Leveraging the usability and extensibility of ModuLM, we can implement and compare a wider range of LLM-based MRL models, thereby gaining insights into how model design impacts performance.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: **[Yes]**

Justification: In this paper, we propose a model framework that supports faster and easier construction of LLM-based molecular representation learning (MRL) models, and we outline our main contributions and scope in the Abstract and Introduction sections (see Abstract and Introduction).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: **[Yes]**

Justification: In this work, we systematically discuss the limitations of our research and outline directions for future work (See Conclusion Section).

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include experimental results related to theoretical aspects.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the code necessary for replicating the studies described in this paper via an anonymous link, and we detail the experimental setup for the replication in the article itself (See Experiments and Appendix).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: For the datasets disclosed in the article, we have provided information regarding their sources and origins (See Appendix).

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: we have specified all the training and test details (e.g., data splits, hyperparameters) necessary to understand the results (See Appendix).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: In this paper, we have reported the standard deviation of the experiments (See Experiments).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: In this paper, we provide detailed information about the experimental resources, including GPU configurations used in our studies (See Experiments).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: The study presented in this paper conforms to the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: We have provided the societal impacts of the work (See Introduction and Conclusion).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not address issues related to this aspect.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All creators and original owners of the assets used in our paper, such as code, data, and models, have been properly credited. We have explicitly mentioned the licenses and terms of use for each asset and have ensured full compliance with these terms throughout our research.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The research presented in this paper is not concerned with new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: This paper does not involve experiments or research related to human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: This paper does not address potential risks incurred by study participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This paper provides a detailed description of the usage of LLMs in this work (See ModuLM section).

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.