Mol-LLM: Multimodal Generalist Molecular LLM with Improved Graph Utilization

Chanhui Lee[1], **Hanbum Ko**[1], **Yuheon Song**[2], **Yongjun Jeong**[1]
Rodrigo Hormazabal[3,4], **Sehui Han**[4], **Kyunghoon Bae**[4], **Sungbin Lim**[4,5 *], **Sungwoong Kim**[1*]
[1]Department of Artificial Intelligence, Korea University
[2]Department of Artificial Intelligence, UNIST
[3]Kim Jaechul Graduate School of AI, KAIST
[4]LG AI Research
[5]Department of Statistics, Korea University

Abstract

Recent advances in large language models (LLMs) have led to models that tackle diverse molecular tasks, such as chemical reaction prediction and molecular property prediction. Large-scale molecular instruction-tuning datasets have enabled sequence-only (e.g., SMILES or SELFIES) generalist molecular LLMs, and researchers are now exploring multimodal approaches that incorporate molecular structural information for further gains. However, a genuinely multimodal, generalist LLM that covers a broad spectrum of molecular tasks has yet to be fully investigated. We observe that naive next token prediction training ignores graph-structural information, limiting an LLM's ability to exploit molecular graphs. To address this, we propose (i) Molecular structure Preference Optimization (MolPO), which facilitates graph usage by optimizing preferences between pairs of correct and perturbed molecular structures, and (ii) an advanced graph encoder with a tailored pre-training strategy to improve the effect of graph utilization by MolPO. Building on these contributions, we introduce Mol-LLM, the first multimodal generalist model that (a) handles a broad spectrum of molecular tasks among molecular LLMs, (b) explicitly leverages molecular-structure information, and (c) takes advantage of extensive instruction tuning. Mol-LLM attains state-of-the-art or comparable results across the most comprehensive molecular-LLM benchmark—even on out-of-distribution datasets for reaction and property prediction, where it surpasses prior generalist molecular LLMs by a large margin.[2]

1 Introduction

Large language models (LLMs) [1–4] have been widely used to tackle diverse tasks across multiple domains, such as mathematics and code generation, by leveraging their broad knowledge base. This achievement has recently motivated interest in applying LLMs to diverse molecular tasks—including molecular property prediction, chemical reaction prediction, description-guided molecule generation, and molecule captioning—all of which are essential in drug discovery and materials science [5–12]. In particular, most molecular LLMs tend to leverage only one of the two key components for improved molecular language modeling, either molecular structure information or multitask instruction-tuning, rather than combining both. Several studies [8, 10, 11] have moved away from conventional molecular language modeling based on 1D sequence such as SMILES [13] or SELFIES [14], and have instead developed multimodal LLMs that incorporate 2D molecular graphs as an additional input modality, thereby representing molecular structures and topologies more faithfully while achieving

*Corresponding Authors. {sungbin, swkim01}@korea.ac.kr.
[2]The model, code, and data will be publicly available.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

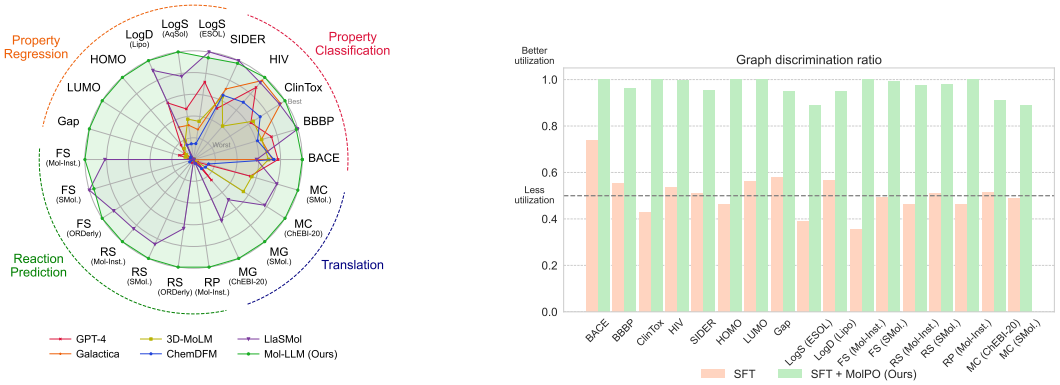

Figure 1: (Left) Performance comparison among generalist molecular LLMs with normalized primary metrics. (Right) Graph utilization comparison between SFT and proposed multimodal training (MolPO). Score closer to 1 indicate better use of graph, approaching 0.5 indicate less utilization.

better performance across diverse molecular tasks [15–17]. Meanwhile, other studies [5–7, 9] have constructed instruction-tuning datasets for multiple molecular tasks and fine-tuned LLMs on these datasets. This approach enables the models to acquire transferable and generalizable knowledge, allowing them to understand and perform various tasks based on natural language instructions.

However, it is uncertain whether the multimodal molecular LLMs effectively use molecular structural information when trained with naive supervised fine-tuning (SFT). To investigate this, we compare the likelihoods of the original and perturbed molecules, comparing how well the SFT model is at proper graph discrimination. Figure 1 shows that the SFT model hardly distinguishes between them on most molecular tasks, indicating that its molecular graph utilization is generally limited. Moreover, despite the potential for synergistic performance improvements by molecular graph structure utilization and multitask instruction-tuning, few studies have fully harnessed the benefits of both approaches, especially for a universal molecular LLM. Specifically, some recent studies [9, 11, 12, 18, 19] have attempted to combine molecule graph structure information with instruction-tuning, however, their instruction-tuning focuses solely on task-specific fine-tuning.

In this paper, we propose a generalist molecular LLM, called Mol-LLM, that leverages multimodal molecule and extensive instruction-tuning, addressing the broadest range of molecular tasks. In particular, while maintaining multimodal LLM architecture based on Q-Former [20], we introduce a novel multimodal instruction-tuning based on Molecular structure Preference Optimization (MolPO), where the molecular LLM learns to optimize the molecular structural preferences between the pairs of the correct (chosen) molecular graph and the perturbed (rejected) molecular graph. By creating rejected molecular graphs based on the substructures for molecular feature perturbation, the proposed MolPO mitigates the tendency to overlook graph information on various molecular tasks. Additionally, to further increase the effect of molecular graph utilization by advanced representation on a wide variety of molecular distributions, we introduce a new graph neural network (GNN) pre-training strategy and architecture. The proposed GNN pre-training framework combines two objectives: (i) functional group prediction, which teaches the model to accurately distinguish functional groups—the features that largely determine molecular properties—and (ii) SELFIES reconstruction, which helps the model preserve the molecular structure details from the molecular graph. Upon GINE [21], adopted by prior multimodal molecular LLMs [8, 9], we incorporate a transformer-based GNN named TokenGT [22], to enhance the expressive power. The resulting Mol-LLM shows strong performance and demonstrably better graph utilization on our benchmarks across a broad range of molecular tasks. To the best of our knowledge, Mol-LLM is not only the first versatile generalist multimodal molecular LLM on a wide range tasks with a single generalist model, but it also surpasses other generalist models: LlaSMol [5], ChemDFM [23], 3D-MoLM [12] on most bencharks as shown in Figure 1, highlighting the power of graph modality synergized with extensive instruction-tuning.

In summary, our contributions are:

1. **Mol-LLM**. We present Mol-LLM, which sets a new state-of-the-art on both in-distribution and out-of-distribution molecular benchmarks relative to existing generalist models.

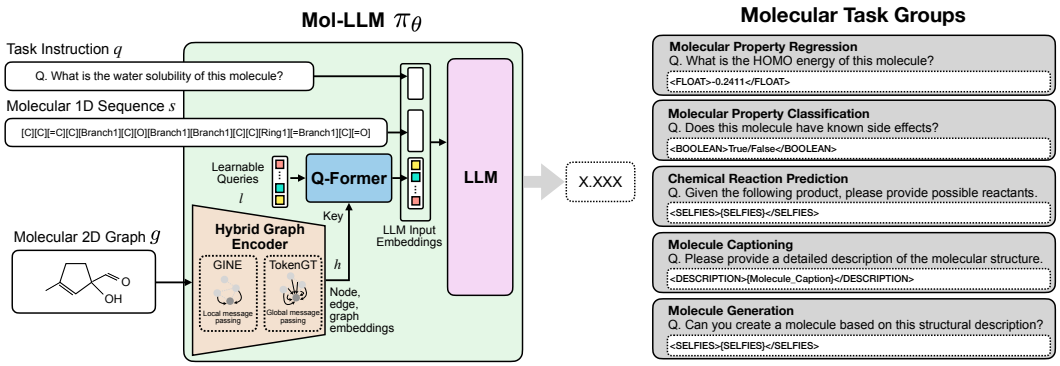

Figure 2: (Left) Overall structure of Mol-LLM. Molecular graph is encoded into a fixed-length token sequence by a hybrid graph encoder, followed by a Q-Former that outputs query embeddings to feed LLM, with corresponding task instruction and molecular 1D sequence. (Right) Representative downstream molecular tasks.

2. **Enhanced graph utilization**. To exploit 2D molecular graphs more effectively, we propose MolPO—a fine-tuning strategy that leverages perturbed molecules—alongside a GNN pre-training method and a hybrid graph encoder augmented with transformer architecture.
3. **Extensive instruction-tuning**. We construct a large, molecule-focused instruction-tuning dataset and employ multimodal training to build a generalist model with significantly enhanced molecular understanding.

2 Mol-LLM: Multimodal Generalist Molecular Large Language Model

This section introduces the model architecture, training strategy, and instruction-tuning dataset of Mol-LLM, a multimodal generalist molecular LLM. As depicted in Figure 2, Mol-LLM comprises a hybrid molecular graph encoder, a Q-Former for cross-modal projection between molecular graph and text, and a backbone LLM. Utilizing the multimodal framework, the LLM addresses molecular task instructions and 1D molecular sequences directly, while feeding 2D molecular graph embeddings to the LLM through the hybrid graph encoder and Q-Former. Such multimodal architectures are trained through three training stages, as depicted in Figure 3.

2.1 Model Architecture

Hybrid Graph Encoder Previous studies on multimodal molecular LLMs using 2D molecular graphs [8, 9] have adopted the GINE architecture [21], since it captures local graphical structure efficiently. However, addressing diverse molecular tasks across various data distributions requires the ability to process large molecules as well. This consideration led us to the simultaneous usage of TokenGT [22] as a graph encoder, which is designed to enhance global context understanding and mitigate over-smoothing [24] in large graphs via a transformer architecture. For a 2D molecular graph $G = (V, E)$, the GINE encoder f^G outputs a graph embedding $h_g^G \in \mathbb{R}^{1\times d_g}$ and node embeddings $h_v^G \in \mathbb{R}^{|V|\times d_g}$, where d_g is the embedding dimension. Otherwise, the TokenGT encoder f^T outputs not only a graph embedding $h_g^T \in \mathbb{R}^{1\times d_g}$ and node embeddings $h_v^T \in \mathbb{R}^{|V|\times d_g}$, but also edge embeddings $h_e^T \in \mathbb{R}^{|E|\times d_g}$. We then concatenate all the embeddings obtained by both encoders $h_g^G, h_v^G, h_g^T, h_v^T$, and h_e^T along the first dimension to obtain $h \in \mathbb{R}^{(2|V|+|E|+2)\times d_g}$, which is then used as the key for the Q-Former.

Cross-modal Projector (Q-Former) Querying transformer (Q-Former) [20] is a modality-bridging transformer that converts the varying number of concatenated embeddings for each molecular graph into a fixed-length token sequence, enabling efficient batch processing. Specifically, structural information is distilled via cross-attention between $N_q = 32$ learnable query vectors $l \in \mathbb{R}^{32\times d_q}$, initialized randomly, and the concatenated molecular embeddings $h \in \mathbb{R}^{(2|V|+|E|+2)\times d_g}$, producing 32 tokens aligned with the text modality. The 32 tokens are concatenated with the task instruction as well as the SELFIES string before being fed to the LLM.

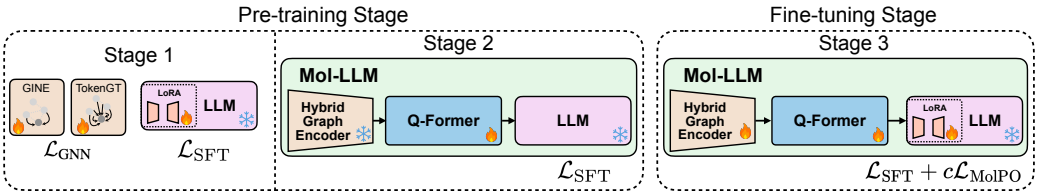

Figure 3: Overview of the three training stages and the loss function used at each stage. The training pipeline consists of a pre-training phase (Stages 1 and 2), followed by a fine-tuning phase (Stage 3). In Stage 1, all modules are trained independently and in parallel, whereas in Stages 2 and 3, the modules are trained in a unified architecture and loss function.

Backbone Large Language Model We adopt Mistral-7B-Instruct-v0.3 [25] as a backbone LLM, following Yu et al. [5]. In order to improve the efficiency in solving molecular tasks, we extend the token codebook with the 3K SELFIES vocabulary from BioT5+ [6] and add dedicated tokens for the digits 0–9, the decimal point, and the negative sign, thereby enabling direct number prediction for regression tasks. Additional task-specific vocabulary covers boolean labels, textual descriptions, and reaction routes, allowing Mol-LLM to natively produce the heterogeneous answer formats required by downstream applications. Examples of these extra tokens appear on the right side of Figure 2.

2.2 Multimodal Training

Stage 1 - Graph Encoder and LLM Pre-training The hybrid graph encoder comprises two GNNs, GINE and TokenGT. We pre-train these two GNNs in parallel with the LLM. The GNN pre-training comprises two complementary tasks: functional group prediction and SELFIES reconstruction. Functional group prediction strengthens representations of the functional groups that govern molecular properties, whereas SELFIES reconstruction encourages the encoder to preserve global structural information. Both tasks share the same graph-level embedding h_g produced by the GNN, as illustrated on the left side of Figure 4. After discarding extremely common or extremely rare functional groups, we retain $K = 72$ distinct groups (dataset construction details are given in Appendix C.1). For functional group prediction, h_g is passed through a three-layer MLP $(1024 \to 1024 \to 72)$ f_θ^{MLP} and trained with the binary cross-entropy loss $\mathcal{L}_{\text{func}} = -\sum_{k=1}^{K}\Big(y_{\text{func}}^{(k)} \log f_\theta^{\text{MLP}}(h_g)^{(k)} + \big(1 - y_{\text{func}}^{(k)}\big) \log\big(1 - f_\theta^{\text{MLP}}(h_g)^{(k)}\big)\Big)$, where superscript (k) is the value for functional group k. SELFIES reconstruction reuses h_g as a context for a GPT-2 decoder $\pi_\theta^{\text{GPT-2}}$ that learns to reproduce molecule's SELFIES string s: $\mathcal{L}_{\text{recon}} = -\sum_t \log \pi_\theta^{\text{GPT-2}}\big(s_t \mid h_g,\, s_{<t}\big)$. The graph encoder is optimized with the combined loss $\mathcal{L}_{\text{GNN}} = \mathcal{L}_{\text{func}} + \mathcal{L}_{\text{recon}}$. Additional training procedures and hyperparameters are provided in Appendix C.2.

The LLM pre-training serves two purposes: (i) injecting molecule-specific prior knowledge and (ii) reducing the compute required during later multimodal training. Accordingly, we pre-train the LLM on exactly the same dataset that will be used later for fine-tuning, optimizing a token-level cross-entropy objective. Given a training instance consisting of a task instruction q, a molecular SELFIES string s, and a ground truth answer y, we minimize $\mathcal{L}_{\text{SFT}} = -\sum_t \log \pi_\theta^{\text{LLM}}\big(y_t \mid s, q, y_{<t}\big)$ where t indexes tokens.

Stage 2 - Q-Former Pre-training In Stage 2, only the Q-Former is updated, while both the GNN and the LLM remain frozen. Following Liu et al. [26], we simply reuse the fine-tuning dataset, in which molecular representations and natural language tokens appear in an interleaved format. For each training instance (s, q, y), the SELFIES string s is converted into its corresponding molecular graph g. The combined model π_θ (GNN+Q-Former+LLM) is then trained for one epoch with the loss defined as $\mathcal{L}_{\text{SFT}} = -\sum_t \log \pi_\theta\big(y_t \mid s, q, g, y_{<t}\big)$.

Stage 3 - MolPO: Molecular Structure Preference Optimization We observed that using only SFT training as in conventional multimodal Molecular LLMs [8, 9, 19, 12], result in a graph bypass phenomenon (Figure 1) in solving molecular tasks. To resolve the graph bypass issue, we propose Molecular structure Preference Optimization (MolPO). Rather than simply inputting multimodal molecules into the LLM without consideration for multimodal utilization, MolPO promotes the practical utilization of multimodal molecules by learning the preferences between an original (chosen)

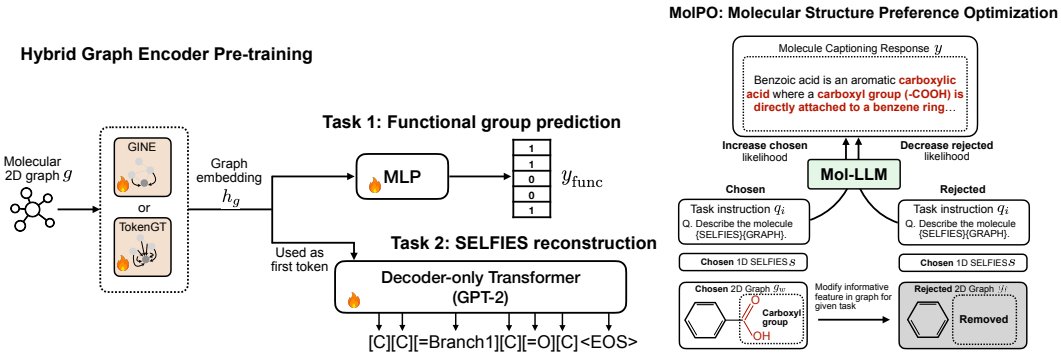

Figure 4: (Left) Overview of the two graph pre-training tasks for the proposed hybrid graph encoder. Two distinct GNN backbones, GINE and TokenGT, are trained independently. (Right) Illustration of the MolPO training objective, which contrasts a chosen molecule with a rejected molecule.

graph g_w and a perturbed (rejected) graph g_ℓ, which is inspired by mDPO [27]. Constructing g_ℓ, it is crucial to introduce perturbations that alter the relationship between the graph and the target. Since molecular features can generally be identified on a substructure basis, we substitute the substructures found in the original g_w as in Figure 4 right panel. This approach offers the advantage of being applicable to any molecular task without significant computational costs or requiring task-specific graph perturbation design (details in Appendix C.4). Based on the reward formulation $r_{w,i} = \frac{\beta}{|y|}\sum_t \log \pi_\theta(y_t \mid g_w, s, q_i, y_{<t})$ and $r_{\ell,i} = \frac{\beta}{|y|}\sum_t \log \pi_\theta(y_t \mid g_\ell, s, q_i, y_{<t})$ motivated from SimPO [28], the MolPO objective is defined as follows:

$$\mathcal{L}_{\text{MolPO}} = \mathbb{E}_{(s,q_i,g,y)\sim\mathcal{D}_{\text{tr}}}[-\log\sigma\big(\min(r_{w,i} - r_{\ell,i}, \lambda_{\text{clip}}|r_{w,i}|) - \gamma_i\big)], \tag{1}$$

where λ_{clip} is coefficient, and $\mathcal{D}_{\text{tr}}$ denotes training dataset. $\gamma_i = \lambda_{\text{margin}}|\mathbb{E}_{(g_w,s,q_i,y)}[r_{w,i}]|$ is a task-adaptive target reward margin for each i-th molecular task, calculated during training. The entire training objective combined is $\mathcal{L}_{\text{SFT}} + c\mathcal{L}_{\text{MolPO}}$, where c is a constant.

In developing a generalist over diverse molecular tasks, we experimentally observed that adopting SimPO's task-agnostic target reward margin γ results in inappropriate log-sigmoid values due to highly variant reward orders of magnitude across tasks. However, modeling the task-specific reward scale as a hyperparameter is not an ideal solution either, as it adds an additional burden of hyperparameter search for each molecular task. Instead, we introduce a task-adaptive target reward margin with only a task-agnostic hyperparameter λ_{margin}, where the expectation is estimated using an exponential moving average during training.

In addition, given that it is generally easier to lower the rejected reward than to increase the true chosen reward, the preference reward margin can be empirically manipulated by simply reducing $r_{\ell,i}$ without a corresponding enhancement of $r_{w,i}$. To fully harness the benefits of preference optimization without the drawbacks, we introduce margin clipping to appropriately control the influence of the reward margin on parameter updates. Specifically, the margin is constrained so that it cannot exceed a fraction, λ_{clip} of $|r_{w,i}|$. Through this simple margin clipping, the model is prevented from the circumventing unintended effect of preference optimization by solely reducing the rejected reward. Further details of MolPO training are provided in Appendix C.5.

2.3 Extensive Instruction-tuning Dataset

The instruction-tuning dataset for Mol-LLM spans five major molecular task groups: property regression, property classification, reaction prediction, description-guided molecule generation, and molecule captioning. Property regression consists of five tasks—LogS for water solubility (ESOL [5]), LogD for lipophilicity (Lipo [5]), HOMO [7], LUMO [7], and HOMO-LUMO gap [7], and property classification comprises BACE [6], BBBP [5], ClinTox [5], HIV [5], SIDER [5]. Reaction prediction covers forward synthesis (FS), retrosynthesis (RS), and reagent prediction (RP), with FS and RS each divided into Mol-Instructions [7] and SMolInstruct [5] subsets according to their dataset sources. The description-guided molecule generation and molecule captioning tasks are similarly split into ChEBI-20 [29] and SMolInstruct based on their origins. In addition, to enhance the understanding of IUPAC [30]—frequently used in molecular text captions—we incorporate an

Table 1: Performance comparison on molecular property prediction tasks from the MoleculeNet [41] benchmark. A superscript * indicates results evaluated with an official checkpoint, and "NA" denotes cases where no official checkpoint is available. **Boldface** highlights the best scores among generalist models. For semi-generalist models, each variant is annotated with the task group on which it is trained. GPT-4 is evaluated with 5-shots, except for classification performances borrowed from Zhao et al. [23] with zero-shot.

Task	LogS	LogD	HOMO	LUMO	Gap	BACE	BBBP	ClinTox	HIV	SIDER
Metric	RMSE (↓)	RMSE (↓)	MAE (↓)	MAE (↓)	MAE (↓)	ROC-AUC (↑)	ROC-AUC (↑)	ROC-AUC (↑)	ROC-AUC (↑)	ROC-AUC (↑)
Specialist Models										
InstructMol	NA	NA	0.0048	0.0050	0.0061	82.1	72.4	NA	68.9	NA
MolCA	≥100	≥100	≥1	≥1	≥1	79.8	70.0	89.5	47.0	63.0
MolXPT	NA	NA	NA	NA	NA	88.4	80.0	95.3	78.1	71.7
Semi-Generalist Models										
Mol-Instructions*	4.81	≥100	0.0210	0.0210	0.0203	41.7	58.0	47.8	49.2	48.2
BioT5+*(Cls. & Trans.)	≥100	≥100	≥1	≥1	≥1	81.1	65.1	83.7	67.0	43.7
BioT5+*(Reg. & React.)	≥100	≥100	0.0022	0.0024	0.0028	65.5	51.5	51.0	58.8	52.5
Generalist Models										
GPT-4 (5-shot)	1.68	1.59	0.0227	0.0462	0.0395	62.5	61.5	51.6	65.9	40.5
Galactica	4.34	2.78	0.2329	0.0413	0.2497	58.4	53.5	78.4	72.2	55.9
3D-MoLM*	3.41	4.86	0.0299	0.0536	0.0673	55.5	53.8	53.7	30.6	49.7
ChemDFM*	8.19	6.21	0.1204	0.1262	0.1694	59.5	50.5	60.0	52.4	51.0
LlaSMol*	**1.21**	1.01	≥1	≥1	≥1	46.7	82.4	77.5	70.3	**78.4**
Mol-LLM (w/o Graph)	1.36	0.95	0.0044	0.0043	0.0055	**80.8**	**84.3**	**85.0**	**76.5**	76.1
Mol-LLM	1.28	**0.91**	**0.0044**	**0.0043**	**0.0054**	80.5	81.1	82.4	75.1	76.3

IUPAC and SELFIES translation dataset [5] to construct an 3.3M extensive instruction-tuning dataset (details in Appendix D.1).

3 Experiments

3.1 Experimental Setup

Baseline Models We group the molecular LLMs compared with Mol-LLM into three broad categories. Specialist models are trained for a single molecular task; semi-generalist models cover a specific task group within one model but do not span all task groups; and generalist models are designed to handle every molecular task group. Representative examples are MolCA [8] for the specialist category, BioT5+ [6] for the semi-generalist category, and Galactica [31] and LlaSMol [5] for the generalist category. Comprehensive details on all baseline models can be found in Appendix E.2.

Evaluation Benchmark In addition to the molecular tasks described in Section 2.3, we evaluate molecular LLM robustness to out-of-distribution (OOD) by proposing two evaluation benchmarks. For LogS prediction, we retain high-confidence solubility labels from AqSol [32], exclude every molecule that also appears in ESOL, and collect molecules of high consistency among labels to construct the OOD evaluation versus ESOL. For reaction prediction, we gather 23K FS and 59K RS data instances from the ORDerly [33] repository except USPTO [34], apply a scaffold split to remove motif overlap with Mol-Instructions [7] and SMolInstruct [5], and reserve 5K examples for evaluation in each task. Full OOD dataset construction details are provided in Appendix D.2.

Evaluation Metrics For property prediction tasks, we report the root mean squared error (RMSE) or mean absolute error (MAE) in regression, and in classification tasks, receiver operating characteristic area under the curve (ROC-AUC) using the predicted probability of the positive class (i.e., *True* token). For reaction prediction and description-guided molecule generation, we evaluate exact match with the target molecule (EXACT), textual similarity (BLEU) [35], molecular fingerprint similarity based on RDKit [36], MACCS keys [37], and Morgan [38] fingerprints (RDK FTS, MACCS FTS, and MORGAN FTS, respectively), and the proportion of generated molecules that are chemically valid (VALIDITY). For the molecule captioning task, we measure similarity between the generated and reference descriptions using BLEU-2, BLEU-4, ROUGE-1 [39], ROUGE-2, ROUGE-L, and METEOR [40]. However, due to space limitations, the main paper reports only the primary metrics. The complete results are provided in Appendix E.3).

3.2 Results

We report the experimental results on property regression and classification, and reaction prediction tasks. In addition, analysis of on molecule captioning and description-guided molecule generation are illustrated in Appendix B.

Table 2: Performance comparison for reaction prediction tasks on Mol-Instructions [7] and SMolInstruct [5] datasets.

Dataset	Mol-Instructions / SMolInstruct					
Task	Forward Synthesis		Retrosynthesis		Reagent Prediction	
Metric	EXACT (↑)	MACCS FTS (↑)	EXACT (↑)	MACCS FTS (↑)	EXACT (↑)	MACCS FTS (↑)
Specialist Models						
InstructMol	0.536 / NA	0.878 / NA	0.407 / NA	0.852 / NA	0.129	0.539
MolCA*	0.000 / 0.000	0.494 / 0.357	0.000 / 0.000	0.880 / 0.760	0.000	0.115
Semi-Generalist Models						
Mol-Instructions*	0.052 / 0.003	0.291 / 0.184	0.069 / 0.015	0.359 / 0.285	0.044	0.364
BioT5+*(Cls. & Trans.)	0.000 / 0.000	0.152 / 0.187	0.001 / 0.000	0.195 / 0.170	0.000	0.056
BioT5+*(Reg. & React.)	0.864 / 0.081	0.975 / 0.537	0.642 / 0.152	0.930 / 0.751	0.257	0.621
Generalist Models						
GPT-4 (5-shot)	0.021 / 0.011	0.728 / 0.634	0.012 / 0.013	0.716 / 0.686	0.000	0.228
Galactica	0.000 / 0.000	0.257 / 0.377	0.000 / 0.000	0.274 / 0.447	0.000	0.127
3D-MoLM*	0.000 / 0.000	0.391 / 0.296	0.000 / 0.000	0.451 / 0.372	0.000	0.218
ChemDFM*	0.000 / 0.002	0.142 / 0.178	0.000 / 0.000	0.440 / 0.443	0.000	0.099
LlaSMol*	0.743 / **0.629**	0.955 / **0.919**	0.453 / 0.323	0.885 / 0.827	0.000	0.199
Mol-LLM (w/o Graph)	0.893 / 0.584	0.983 / 0.904	0.510 / 0.363	0.886 / 0.828	0.202	0.586
Mol-LLM	**0.911** / 0.601	**0.987** / 0.908	**0.538 / 0.377**	**0.893 / 0.832**	**0.225**	**0.600**

Table 3: Evaluation of OOD generalization for reaction prediction on the ORDerly dataset, which is non-USPTO, and LogS on the AqSol dataset.

Dataset	AqSol	ORDerly					
Task	LogS	Forward Synthesis			Retrosynthesis		
Metric	RMSE (↓)	EXACT (↑)	MACCS FTS (↑)	VALIDITY (↑)	EXACT (↑)	MACCS FTS (↑)	VALIDITY (↑)
Semi-Generalist Models							
BioT5+*(Reg. & React.)	1.81	0.095	0.628	1.00	0.139	0.678	1.00
Generalist Models							
GPT-4	2.17	0.000	0.723	0.87	0.000	0.672	0.65
Galactica*	3.20	0.000	0.322	0.49	0.000	0.398	0.38
3D-MoLM*	2.72	0.000	0.288	0.01	0.000	0.396	0.01
ChemDFM*	6.98	0.017	0.428	0.04	0.000	0.406	0.05
LlaSMol*	1.32	0.350	0.881	**1.00**	0.473	0.875	0.99
Mol-LLM (w/o Graph)	1.10	0.394	**0.900**	**1.00**	0.727	0.936	**1.00**
Mol-LLM	**1.02**	**0.401**	0.877	**1.00**	**0.738**	**0.939**	**1.00**

Property Regression and Classification Table 1 summarizes the property regression and classification results. On most tasks, Mol-LLM outperforms every other generalist model, except for LogS, ClinTox, and SIDER. Notably, even Mol-LLM (w/o Graph) performs on a par with the full model. We attribute this behavior to the small molecular sizes in MoleculeNet [41], which allow the LLM to infer structural information directly from the SELFIES representation.

Reaction Prediction The reaction prediction results are reported in Table 2. Except for the FS task of SMolInstruct dataset, Mol-LLM again leads all generalist models. Since successful reaction prediction depends on recognizing which functional groups can participate during a chemical reaction, these results suggest that pre-training of the GNN on functional group prediction helps Mol-LLM exploit structural cues more effectively. Consistent with this interpretation, omitting the graph input (w/o Graph variant) noticeably degrades performances.

Generalization Performance on Out-of-distribution Datasets Table 3 reports OOD results for AqSol. On the in-distribution training tasks (LogS and SIDER), Mol-LLM lags the generalist baseline LlaSMol only marginally. In contrast, it is markedly superior on the OOD AqSol benchmark, demonstrating stronger generalization. A similar trend appears in the reaction prediction FS and RS tasks: Mol-LLM is slightly weaker on in-distribution FS of SMolInstruct but outperforms competitors when evaluated OOD. These findings indicate that MolPO training confers broader generalization across both tasks and input distributions, whereas the semi-generalist BioT5+, which lacks large-scale instruction tuning, suffers a notable drop in performance.

Table 4: An ablation study on MolPO's effect on graph utilization. We report RMSE(↓) for LogS and LogD, and EXACT(↑) for FS, RS, RP, and T2M, each representing forward reaction prediction, retrosynthesis, reagent prediction, and molecule generation. "Mol-Inst." and "SMol." denote the Mol-Instructions and SMolInstruct datasets, respectively.

	LogS	LogD	FS (Mol-Inst.)	FS (SMol.)	RS (Mol-Inst.)	RS (SMol.)	RP (Mol-Inst.)	T2M (ChEBI-20)	T2M (SMol.)
Mol-LLM (w/o MolPO)	1.36	0.96	0.907	0.598	0.529	0.368	0.220	0.426	0.355
Mol-LLM	**1.28**	**0.91**	**0.911**	**0.601**	**0.538**	**0.377**	**0.225**	**0.443**	**0.368**

3.3 Ablation Study

MolPO objective enhances molecular graph utilization and task performance. To examine whether incorporating the MolPO objective $\mathcal{L}_{\text{MolPO}}$ during Mol-LLM training leads the model to exploit molecular graph information more effectively than training with SFT alone, we first compare, for each task i, the log-likelihood $r_{w,i}$ obtained when the model is given the chosen graph g_w to the log-likelihood $r_{\ell,i}$ obtained when it is given the rejected graph g_ℓ. We then compute the graph discrimination ratio GDR $= \frac{1}{N_i}\sum_{n=1}^{N_i} \mathbb{I}\big[r_{w,i}(n) > r_{\ell,i}(n)\big]$, where N_i is the number of instances in task i, $\mathbb{I}$ is the indicator function, and $r_{w,i}(n)$ is the log-likelihood for the n-th instance in task i. A GDR close to 1 indicates that the model can clearly identify the correct molecular graph (i.e., it effectively exploits molecular graph information); a value near 0.5 indicates random guessing, and a value near 0 indicates systematic confusion. Figure 1 shows the per-task GDRs—green bars for MolPO-trained models and orange bars for models trained without $\mathcal{L}_{\text{MolPO}}$. The consistently higher GDRs in the MolPO setting confirm that this objective helps the model make better use of molecular graph information. We also compare the multitask fine-tuning performance obtained when $\mathcal{L}_{\text{MolPO}}$ is combined with $\mathcal{L}_{\text{SFT}}$ to that obtained when only $\mathcal{L}_{\text{SFT}}$ is used. As shown in Table 4, leveraging the graph modality through the MolPO objective improves performances on most tasks.

Furthermore, we demonstrate the effectiveness of the our GNN pre-training in Appendix B.

4 Related Works

Molecular Large Language Models MolT5 [29] extends T5 [42] to bidirectional translation between SMILES strings and natural language, whereas MolXPT [43], built on the GPT architecture [44], unifies text–molecule translation with property prediction. MolCA [8] and GIT-Mol [10] fuse 2D molecular graphs with text via a Q-Former [20], while MolLM [45] further injects 3D geometric cues. UniMoT [11] discretizes Q-Former outputs into graph tokens while 3D-MolT5 [19] introduces 3D structure tokens, enabling generative reasoning over conformers. Although these models exploit molecule structures, each is tailored to a narrow set of tasks. Mol-LLM tackles this limitation by jointly processing text and graphs and by performing translation, prediction, and generation within a single generalist framework.

Instruction-tuning on Molecular Tasks Mol-Instructions [7] introduced the first broad instruction-tuning corpus, inspiring InstructMol [9] to fine-tune multimodal models with task-specific prompts and BioT5+ [6] to perform multitask tuning without structural inputs. LlaSMol [5] scales the idea to 3.3M examples across ten tasks, yielding a single model that matches—or exceeds—specialists. Subsequent work, including UniMoT [11], 3D-MolT5 [19] and 3D-MoLM, couples instruction tuning with 2D/3D structure encoders, yet still lacks a systematic strategy for exploiting multimodal inputs. Consequently, models remain sensitive to task distribution shifts. Mol-LLM fills this gap by unifying instruction tuning with structure-aware training, thereby improving robustness across in-distribution and out-of-distribution tasks.

Preference Optimization on Different Modality DPO [46] aligns language models with human preferences by maximizing the log-probability gap between preferred and rejected outputs; SimPO [28] removes the expensive reference model for lighter training. As multimodal LLMs rise, mDPO [27] adapts the idea to vision–language models by corrupting images to build preference pairs, and numerous follow-ups [47–50] confirm its effectiveness. Yet no study has demonstrated comparable gains for molecular data. Mol-LLM is the first to apply preference optimization to molecular graphs and text jointly, showing that structure-aware preferences yield stronger generalization than sequence-only tuning while keeping training costs manageable.

5 Conclusion

We introduced MolPO, a multimodal training objective that leverages perturbed molecules to enhance the utility of 2D molecular graphs, together with a hybrid graph encoder pre-training strategy. We also curated a large-scale molecule instruction tuning dataset and, using the proposed methods, developed Mol-LLM, a multimodal generalist molecular large language model. Mol-LLM achieved state-of-the-art performances among generalist molecular models on property regression, property classification, reaction prediction, description-guided molecule generation, and molecule captioning tasks. We believe our approach can be extended beyond 2D molecular graphs to incorporate 3D structural information and molecular metadata, enabling real-world applications such as drug discovery and novel material discovery. A detailed discussion of the limitations are described in Appendix A.

Acknowledgments and Disclosure of Funding

LG AI Research supported this work. This work was also supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea)&Gwangju Metropolitan City, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00410082), Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University); No.RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST); No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI), Institute of Information & communications Technology Planning & Evaluation (IITP) under the artificial intelligence star fellowship support program to nurture the best talents (IITP-2025-RS-2025-02304828, 50%) grant funded by the Korea government (MSIT), and partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ITRC (Information Technology Research Center) grant funded by the Korea government (MSIT) (IITP-2025-RS-2024-00436857, 20%).

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2023.

[2] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[3] Gemini Team Google. Gemini: A family of highly capable multimodal models. *ArXiv*, abs/2312.11805, 2023.

[4] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL `https://arxiv.org/abs/2307.09288`.

[5] Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *ArXiv*, abs/2402.09391, 2024. URL `https://api.semanticscholar.org/CorpusID:267657622`.

[6] Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. *ArXiv*, abs/2402.17810, 2024. URL https://api.semanticscholar.org/CorpusID:268041632.

[7] Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *ArXiv*, abs/2306.08018, 2023. URL https://api.semanticscholar.org/CorpusID:259164901.

[8] Zhiyuan Liu, Sihang Li, Yancheng 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*, abs/2310.12798, 2023. URL https://api.semanticscholar.org/CorpusID:264306303.

[9] He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *ArXiv*, abs/2311.16208, 2023. URL https://api.semanticscholar.org/CorpusID:265466509.

[10] Pengfei Liu, Yiming Ren, Jun Tao, and Zhixiang Ren. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *Computers in Biology and Medicine*, 171:108073, March 2024. ISSN 0010-4825. doi: 10.1016/j.compbiomed.2024.108073. URL http://dx.doi.org/10.1016/j.compbiomed.2024.108073.

[11] Juzheng Zhang, Yatao Bian, Yongqiang Chen, and Quanming Yao. Unimot: Unified molecule-text language model with discrete token representation, 2024. URL https://arxiv.org/abs/2408.00863.

[12] 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.

[13] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, feb 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005. URL https://doi.org/10.1021/ci00057a005.

[14] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100*Machine Learning: Science and Technology*, 1(4):045024, October 2020. ISSN 2632-2153. doi: 10.1088/2632-2153/aba947. URL http://dx.doi.org/10.1088/2632-2153/aba947.

[15] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Anima Anandkumar. Multi-modal molecule structure-text model for text-based retrieval and editing, 2024. URL https://arxiv.org/abs/2212.10789.

[16] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4:279–287, 2022.

[17] Bing Su, Dazhao Du, Zhao Yang, Yujie Zhou, Jiangmeng Li, Anyi Rao, Hao Sun, Zhiwu Lu, and Ji-Rong Wen. A molecular multimodal foundation model associating molecule graphs with natural language, 2022. URL https://arxiv.org/abs/2209.05481.

[18] Youwei Liang, Ruiyi Zhang, Li Zhang, and Pengtao Xie. Drugchat: Towards enabling chatgpt-like capabilities on drug molecule graphs, 2023. URL https://arxiv.org/abs/2309.03907.

[19] Qizhi Pei, Lijun Wu, Kaiyuan Gao, Jinhua Zhu, and Rui Yan. 3d-molt5: Towards unified 3d molecule-text modeling with 3d molecular tokenization, 2024. URL https://arxiv.org/abs/2406.05797.

[20] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

[21] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

[22] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems*, 35:14582–14595, 2022.

[23] Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, et al. Chemdfm: A large language foundation model for chemistry. *arXiv preprint arXiv:2401.14818*, 2024.

[24] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[25] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *ArXiv*, abs/2304.08485, 2023. URL `https://api.semanticscholar.org/CorpusID:258179774`.

[27] Fei Wang, Wenxuan Zhou, James Y. Huang, Nan Xu, Sheng Zhang, Hoifung Poon, and Muhao Chen. mdpo: Conditional preference optimization for multimodal large language models. *ArXiv*, abs/2406.11839, 2024. URL `https://api.semanticscholar.org/CorpusID:270560448`.

[28] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *ArXiv*, abs/2405.14734, 2024. URL `https://api.semanticscholar.org/CorpusID:269983560`.

[29] Carl N. Edwards, T. Lai, Kevin Ros, Garrett Honke, and Heng Ji. Translation between molecules and natural language. *ArXiv*, abs/2204.11817, 2022. URL `https://api.semanticscholar.org/CorpusID:248376906`.

[30] Henri A Favre and Warren H Powell. *Nomenclature of organic chemistry: IUPAC recommendations and preferred names 2013*. Royal Society of Chemistry, 2013.

[31] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony S. Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *ArXiv*, abs/2211.09085, 2022. URL `https://api.semanticscholar.org/CorpusID:253553203`.

[32] Murat Cihan Sorkun, Abhishek Khetan, and Süleyman Er. Aqsoldb, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. *Scientific Data*, 6, 2019. URL `https://api.semanticscholar.org/CorpusID:199491456`.

[33] Daniel S. Wigh, Joe Arrowsmith, Alexander Pomberger, Kobi C. Felton, and Alexei A. Lapkin. Orderly: Data sets and benchmarks for chemical reaction data. *Journal of Chemical Information and Modeling*, 64:3790 – 3798, 2024. URL `https://api.semanticscholar.org/CorpusID:269325115`.

[34] Jinmao Wei, Xiao-Jie Yuan, Qinghua Hu, and Shuqin Wang. A novel measure for evaluating classifiers. *Expert Syst. Appl.*, 37:3799–3809, 2010. URL `https://api.semanticscholar.org/CorpusID:9240275`.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[36] Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.

[37] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42 (6):1273–1280, 2002.

[38] Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2): 107–113, 1965.

[39] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[40] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[41] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9:513 – 530, 2017. URL `https://api.semanticscholar.org/CorpusID:217680306`.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[43] Zequn Liu, Wei Zhang, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Ming Zhang, and Tie-Yan Liu. Molxpt: Wrapping molecules with text for generative pre-training. In *ACL*, 2023.

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[45] 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. *Bioinformatics*, 40:i357 – i368, 2024. URL `https://api.semanticscholar.org/CorpusID:265455405`.

[46] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf`.

[47] Wenyi Xiao, Ziwei Huang, Leilei Gan, Wanggui He, Haoyuan Li, Zhelun Yu, Hao Jiang, Fei Wu, and Linchao Zhu. Detecting and mitigating hallucination in large vision language models via fine-grained ai feedback. *ArXiv*, abs/2404.14233, 2024. URL `https://api.semanticscholar.org/CorpusID:269293208`.

[48] Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. Aligning modalities in vision large language models via preference fine-tuning. *ArXiv*, abs/2402.11411, 2024. URL `https://api.semanticscholar.org/CorpusID:267750239`.

[49] Renjie Pi, Tianyang Han, Wei Xiong, Jipeng Zhang, Runtao Liu, Rui Pan, and Tong Zhang. Strengthening multimodal large language model with bootstrapped preference optimization. *ArXiv*, abs/2403.08730, 2024. URL `https://api.semanticscholar.org/CorpusID:268379605`.

[50] Yihe Deng, Pan Lu, Fan Yin, Ziniu Hu, Sheng Shen, James Zou, Kai-Wei Chang, and Wei Wang. Enhancing large vision language models with self-training on image comprehension. *ArXiv*, abs/2405.19716, 2024. URL `https://api.semanticscholar.org/CorpusID:270123045`.

[51] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[52] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://api.semanticscholar.org/CorpusID:202558505.

[53] 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.

[54] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Y. Zaslavsky, Jian Zhang, and Evan E. Bolton. Pubchem 2023 update. *Nucleic acids research*, 2022. URL https://api.semanticscholar.org/CorpusID:253182955.

Table 5: Performance comparison for molecule generation and molecule captioning on ChEBI-20 [29] and SMolInstruct [5] datasets.

Dataset	ChEBI-20 / SMolInstruct					
Task	Molecule Generation			Molecule Captioning		
Metric	EXACT (↑)	MACCS FTS (↑)	VALIDITY (↑)	BLEU-4 (↑)	ROUGE-L (↑)	METEOR (↑)
Specialist Models						
GIT-Mol	0.051 / NA	0.738 / NA	0.93 / NA	0.263 / NA	0.560 / NA	0.533 / NA
InstructMol	NA / NA	NA / NA	NA / NA	0.371 / NA	0.502 / NA	0.509 / NA
MolT5	0.311 / 0.317	0.834 / 0.879	0.91 / 0.95	0.508 / 0.366	0.594 / 0.501	0.614 / 0.515
MolCA*	NA / NA	NA / NA	NA / NA	0.540 / 0.510	0.631 / 0.604	0.652 / 0.628
MolXPT	0.215 / NA	0.859 / NA	0.98 / NA	0.505 / NA	0.597 / NA	0.626 / NA
Text+Chem T5	0.322 / NA	0.901 / NA	0.94 / NA	0.542 / NA	0.622 / NA	0.648 / NA
Semi-Generalist Models						
Mol-Instructions	0.016 / 0.045	0.167 / 0.475	1.00 / 1.00	0.171 / 0.020	0.289 / 0.217	0.271 / 0.124
BioT5+*(Cls. & Trans.)	0.557 / 0.519	0.907 / 0.897	1.00 / 1.00	0.591 / 0.582	0.649 / 0.644	0.680 / 0.677
BioT5+*(Reg. & React.)	0.537 / 0.416	0.897 / 0.867	1.00 / 1.00	0.216 / 0.221	0.364 / 0.364	0.323 / 0.321
Generalist Models						
GPT-4 (5-shot)	0.092 / 0.027	0.745 / 0.726	0.65 / 0.74	0.158 / 0.125	0.303 / 0.273	0.320 / 0.274
Galactica*	0.000 / 0.000	0.264 / 0.271	0.70 / 0.61	0.000 / 0.000	0.006 / 0.006	0.004 / 0.005
3D-MoLM*	0.000 / 0.000	0.000 / 0.000	0.00 / 0.00	0.171 / 0.167	0.287 / 0.285	0.326 / 0.329
ChemDFM*	0.018 / 0.041	0.165 / 0.297	0.19 / 0.13	0.031 / 0.035	0.101 / 0.108	0.078 / 0.085
LlaSMol*	0.274 / 0.180	0.871 / 0.845	0.95 / 0.93	0.333 / 0.328	0.464 / 0.465	0.466 / 0.470
Mol-LLM (w/o Graph)	0.431 / 0.362	0.903 / **0.888**	**1.00** / **1.00**	0.482 / 0.477	**0.509** / **0.490**	0.587 / 0.585
Mol-LLM	**0.443** / **0.368**	**0.906** / 0.887	**1.00** / 0.99	**0.493** / **0.482**	0.439 / 0.433	**0.599** / **0.589**

A Limitation

Performance Degradation from Limited Molecular Distribution in Classification Tasks When the training data lacks sufficient diversity, preference optimization approaches using input preference pairs could suffer performance degradation on test or out-of-distribution datasets. In the case of MolPO, if the training molecular distribution is too narrow or contains spurious patterns unrelated to the given molecular task, the model may inappropriately regard molecules in test set or out-of-distribution (OOD) dataset as rejected molecules, based solely on their non-in-distribution characteristics. This hypothesis is consistent with the observations in Table 1 for the classification datasets. The classification datasets are substantially smaller than the datasets in the other task groups. More than half of them contain only approximately 1K samples, compared with 3.3M samples in the entire training dataset, which explains why MolPO's performance either remained unchanged or slightly decreased. The principled and necessary solution to this issue is basically to procure more diverse molecular distributions. We anticipate that the research community will pay more attention to developing diverse and comprehensive property classification datasets.

In-depth Analysis across Molecular Tasks Beyond the overall improvement in benchmark performance, an in-depth analysis is needed to understand what qualitative changes occur for each molecular task from the improved graph utilization by MolPO. It is necessary to identify trends that cannot be determined by performance metrics alone, such as which molecular features are difficult to capture with sequence-only approaches, and whether these identified molecular features have strong practical impact. Such analysis could be particularly interesting for property prediction tasks where spatial recognition of molecules is important.

Multi-step Reasoning and Multi-turn Interaction As demonstrated by recent successful LLMs [2, 3], impactful real-world applications of LLMs critically depend on multi-step reasoning capabilities and multi-turn interactions between LLMs and users. However, research on these two aspects remains significantly underdeveloped in the field of molecular LLMs. Such research requires different considerations from single-turn instruction tuning, beginning with dataset construction, and necessitates appropriate training objectives and reward modeling. It is an interesting direction to extend molecular LLMs to multi-step reasoning and multi-turn interaction for practical applications.

B Additional Experimental Results

Description-guided Molecule Generation Table 5 shows the results for description-guided molecule generation, whose input prompts contain no molecular graphs. Since both Mol-LLM

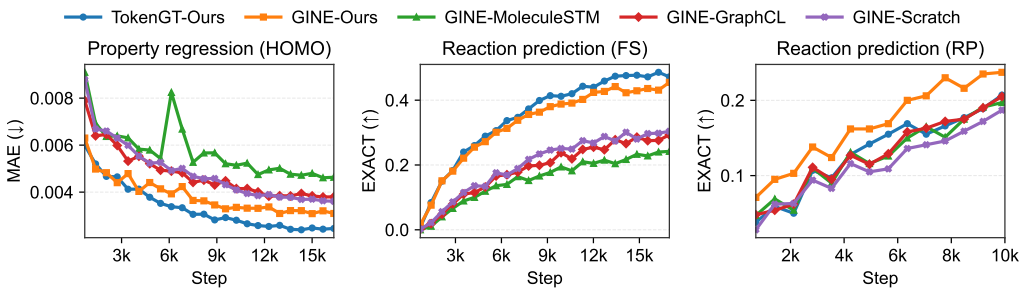

Figure 5: Comparison of fine-tuning performances on three tasks under different GNN architectures and initialization with (or without) pre-trained parameters. Each line is labeled as {GNN architecture}–{initialization}. *Ours* refers to models initialized with parameters obtained via our GNN pre-training method, whereas *Scratch* denotes models trained from random initialization. The x-axis denotes the number of training steps, and the y-axis shows the corresponding evaluation metric.

and the w/o Graph variant receive identical inputs, their scores are nearly indistinguishable. This confirms that Mol-LLM's ability to use graphs does not impede its instruction-following ability when graphs are absent. On both the ChEBI-20 and SMolInstruct datasets, Mol-LLM nonetheless achieves the best results among generalist models.

Molecule Captioning As summarized in Table 5, Mol-LLM again surpasses all baselines. Compared with the w/o Graph variant, the full model obtains consistently higher BLEU and METEOR scores but slightly lower ROUGE scores on both ChEBI-20 and SMolInstruct. The pattern implies that Mol-LLM produces more concise captions: it captures the essential information while omitting peripheral details. We believe MolPO training encourages the model to rely on structural cues and focus on the core content.

Ablation study on Our GNN pre-training To clearly demonstrate the effect of our GNN pre-training method, we frame the experiment as a single task setting and modify Mol-LLM so that, during fine-tuning, it receives only the task instruction and the 2D molecular graph as inputs, omitting the 1D sequence. The model is trained solely using the loss term $\mathcal{L}_{\text{SFT}}$, and its performance is then compared with different GNN architectures and weight initializations. Figure 5 presents the learning curves for property regression (HOMO) and reaction prediction (FS, RP). The GNN architectures (GINE, TokenGT) and their corresponding initializations are represented as {GNN architecture}-{initialization}. *Scratch* indicates that the GNN is trained from scratch without any pre-trained weights. The model whose GNN is initialized with the proposed pre-training method (*Ours*) consistently outperforms the others, indicating that it learns higher quality molecular representations. Moreover, the existing pre-trained models—MoleculeSTM [15] and GraphCL [51]—perform either worse than or roughly on par with the non-pretrained baseline *Scratch*, which is a surprising outcome.

C Implementation Details

This section discusses the details of the Mol-LLM implementation. All the necessary materials to reproduce the results through Tables 1 to 3 and 5, including code, trained model, and test set, are available at `https://anonymous.4open.science/r/mol-llm-neurips2025-93EB`.

C.1 Functional Group Prediction Dataset for Graph Encoder Pre-training

As explained in Section 2.2, the proposed graph encoder pre-training conducts functional group prediction of a given molecule, a kind of self-supervision task carried out only with the input molecule. The principal challenge in constructing the functional group prediction dataset is the severe class imbalance: some groups occur in most molecules, whereas others are exceedingly rare. Leveraging the RDKit Fragments module[3], we enumerate 87 functional groups and quantify their occurrences across the entire PubChem database, as summarized in the top panel of Figure 6. Figure 6 illustrates functional group imbalance, for example, `fr_NH0` (tertiary amines) appears in many molecules,

[3]https://www.rdkit.org/docs/source/rdkit.Chem.Fragments.html

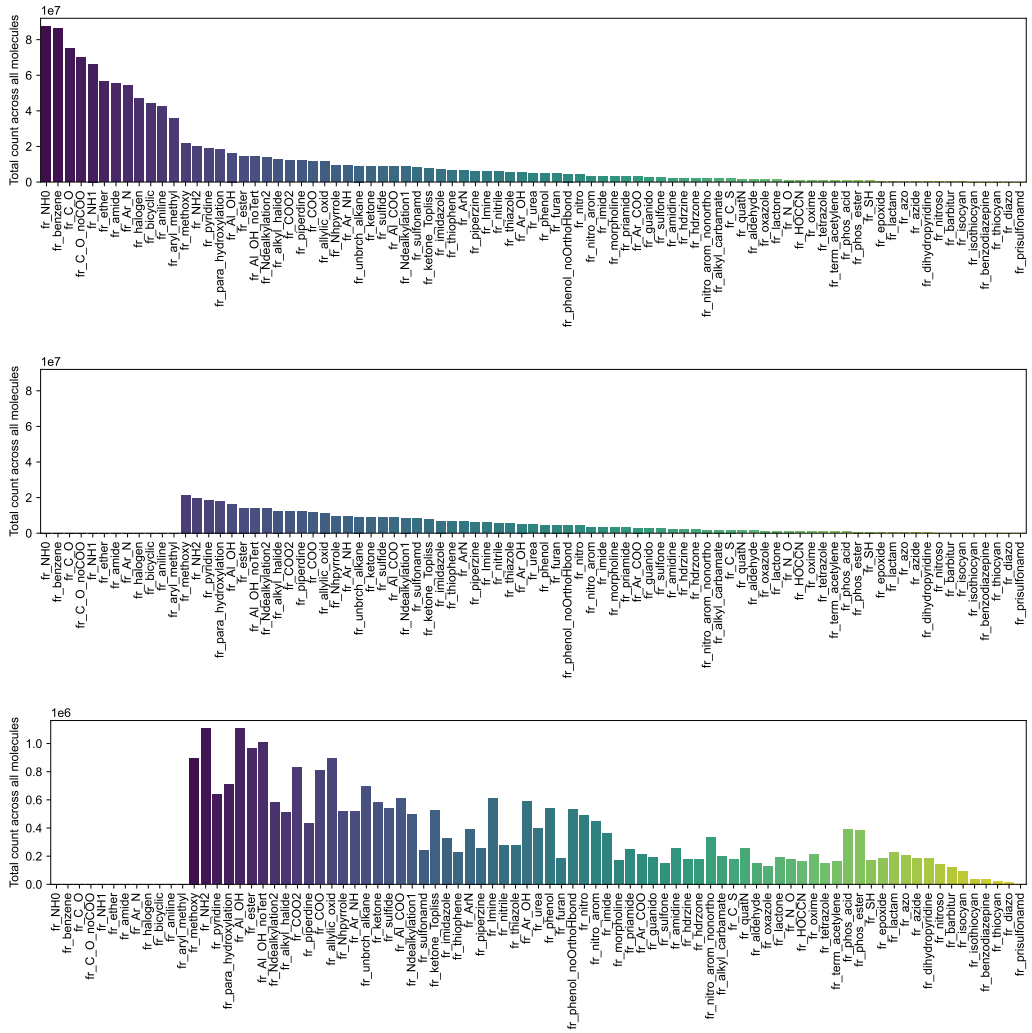

Figure 6: (Top) Distribution of functional groups present in molecules from the PubChem database. (Middle) Distribution of functional groups in PubChem molecules after excluding groups that are either overly common or extremely rare. (Bottom) Distribution of functional groups obtained after sampling 5M molecules from PubChem database, considering functional group sparsity. Since the number of molecules differs among panels, the y-axis scale varies across plots for visualization purposes.

whereas `fr_prisulfonamd` (primary sulfonamides) are scarce. This imbalance can cause overfitting to dominant classes instead of learning general chemical knowledge. To alleviate the overfitting problem, we remove the 11 most prevalent groups (from `fr_NH0` to `fr_aryl_methyl`) and the rarest group (`fr_prisulfonamd`), retaining 72 functional groups. The middle panel of Figure 6 shows the reduced yet still skewed distribution. To adjust the skewed distribution, we apply sparsity-aware importance sampling as follows. Given M molecules and G retained groups, let $x_{i,g} \in \{0, 1\}$ indicate the presence of group g in molecule i. Then we can define group frequencies as $c_g = \sum_{i=1}^{M} x_{i,g}$. Here, we implement importance sampling that favors rarer groups by introducing the scaling factor $s_g = 1/(c_g + \varepsilon)$ with $\varepsilon = 10^{-6}$, resulting in the sparsity score of molecule i as

$$\sigma_i = \Big(\sum_{g=1}^{G} x_{i,g}\, s_g \Big)^2 .$$

Normalizing the scores yields a categorical distribution $p_i = \sigma_i / \sum_{j=1}^{M} \sigma_j$, from which we sample 5M molecules. The resulting distribution (bottom panel of Figure 6) is flatter than before, which

Table 6: Comparison of MAE (↓) across different GNN training settings on the QM9 dataset.

Property	Description	GINE (Tuning)	GINE (Frozen)	GINE (Frozen MoleculeSTM)
μ	Dipole moment	**0.5247**	0.9616	1.0927
α	Isotropic polarizability	**1.026**	3.1919	3.3589
ϵ_{HOMO}	Highest occupied molecular orbital energy (HOMO)	**0.1558**	0.2864	0.357
ϵ_{LUMO}	Lowest unoccupied molecular orbital energy (LUMO)	**0.1428**	0.3555	0.4581
$\Delta\epsilon$	Gap between ϵ_{HOMO} and ϵ_{LUMO} (Gap)	**0.1817**	0.4003	0.3997
$<R^2>$	Electronic spatial extent	**24.7215**	94.5913	103.2612
$ZPVE$	Zero point vibrational energy	**0.0471**	0.3056	0.234
U_0	Internal energy at 0K	**9,474.99**	10,302.67	10,176.77
U	Internal energy at 298.15K	10,160.12	10,550.07	**10,134.84**
H	Enthalpy at 298.15K	10,295.32	10,466.08	**10,057.47**
G	Free energy at 298.15K	**9,596.59**	10,278.72	10,142.24
c_v	Heat capavity at 298.15K	**0.5053**	1.1965	1.4149
U_0^{ATOM}	Atomization energy at 0K	**0.9713**	3.6615	3.2477
U^{ATOM}	Atomization energy at 298.15K	**0.8442**	3.4631	3.309
H^{ATOM}	Atomization enthalpy at 298.15K	**0.999**	3.6308	3.2287
G^{ATOM}	Atomization free energy at 298.15K	**1.0225**	3.5317	3.4369
A	Rotational constant	**0.9253**	0.7283	1.0479
B	Rotational constant	**0.1515**	0.2428	0.2511
C	Rotational constant	**0.0773**	0.172	0.1368

enables the graph encoder to learn more unbiased chemical knowledge than when trained on the raw PubChem molecule distribution.

C.2 Details of Graph Encoder and Pre-training

Architecture Both GNN components of the hybrid graph encoder—GINE and TokenGT—use a hidden dimension $d_g = 1024$ and five message-passing layers. We replace the original transformer blocks of TokenGT with a BERT encoder implemented in `FlashAttention-2` and configured with eight attention heads, thereby maximizing GPU throughput. Within TokenGT, the node- and edge-projection dimensions are both 64, and we adopt the graph laplacian eigenvector variant for node positional encoding.

Pre-training GINE and TokenGT are pre-trained with the same set of hyperparameters. For SELFIES reconstruction, sequences are truncated to a maximum length of 512 tokens; tokens beyond this limit do not contribute to the loss calculation. The GPT-2 decoder used for reconstruction consists of six layers, eight attention heads, and an embedding size of 1024. We train for 50 epochs with a learning rate of 1×10^{-4}, a batch size of 64, and the AdamW optimizer. Training is performed on the 5M molecule dataset described in Appendix C.1, further augmented only by adding the corresponding SELFIES strings, and all experiments are run on four NVIDIA A100 GPUs.

C.3 Investigation of Graph Encoder Used in Prior Work

In Appendix B, we show that the downstream performance of the LLM integrated with pre-trained GNNs used in Liu et al. [8], Cao et al. [9], which are MoleculeSTM [15] and GraphCL [51], does not improve from that of random initialization. To further investigate the graph representation of MoleculeSTM, we conducted an additional experiment evaluating MoleculeSTM in isolation from the LLM on the QM9 datasets. In this experiment, the graph embedding h_g is obtained by mean-pooling the node embeddings, and then passed to a simple MLP, a regression head, whose output is used for training over MSE minimization. While tuning the regression head, we compare three GNN tuning settings: tuning a randomly initialized GNN, freezing a randomly initialized GNN, and freezing a GNN initialized with MoleculeSTM. Table 6 reports the mean absolute error (MAE) for each property. It turns out that, when only the regression head is trained, the gap between random and MoleculeSTM initialization remains negligible w.r.t. the jointly training of GINE, reinforcing our observation that the current pre-trained GNN model fails to capture useful molecular representations. All models were trained for 1,500 epochs with a batch size of 128 using the Adam optimizer with a learning rate of 10^{-4} on four NVIDIA A100 GPUs.

C.4 Molecular Structure Preference Pair

To improve the graph utilization of our model, we create molecular structural preference pairs, which are required for Molecular Structure Preference Optimization (MolPO). Specifically, as a generalist

Table 7: Model hyperparameters used for Mol-LLM architecture, evaluation, and training stages.

(a) Q-Former, LoRA, evaluation

Parameter	Value
Q-Former	
bert_hidden_dim	768
bert_name	`scibert` [52]
num_query_token	32
bert_layers	5
LoRA	
lora_r	64
lora_alpha	32
lora_dropout	0.1
Evaluation	
gen_max_len	256
num_beams	1

(b) Training hyperparameters for each stage

Parameter	Stage 1	Stage 2	Stage 3
max_length		512	
batch_size	968	1024	1024
optimizer		`adamw`	
scheduler		`linear_warmup_cosine_lr`	
weight_decay		0.05	
min_lr		10^{-5}	
init_lr	10^{-4}	10^{-4}	4×10^{-5}
warmup_lr	10^{-5}	10^{-5}	4×10^{-6}
warmup_epochs		0.25	
gradient_clip_val		0.5	
precision		`bf16-mixed`	
c	NA	NA	0.25
λ_{margin}	NA	NA	0.25
λ_{clip}	NA	NA	1.0

molecular LLM, it requires a preference pair generation method applicable across various molecular tasks. Therefore, we employed functional group-based substructure modification, which can alter molecular features based on only the input molecule, without requiring task-specific design. For this, we propose Molecular ACCess System (MACCS) [37] keys-based substructure modification method directly modifies molecular substructures by randomly removing and adding them. This approach first identifies the substructures of the molecule corresponding to MACCS keys, generating two lists: one containing the MACCS keys representing functional groups present in the molecule, and the other containing the keys for functional groups absent from the molecule. Then we sample random keys from the present MACCS keys to remove from the original molecular graph. Subsequently, other random keys are chosen from the list of absent MACCS keys, and functional groups corresponding to the selected MACCS keys are attached at a random position in the molecule. We set the number of MACCS keys randomly selected to 30 percent of the number of each molecule's present MACCS keys. This method effectively alters molecular structural information without task-specific design, at the same time, it does not require heavy computation.

C.5 Details of Mol-LLM

This section describes the details of the Mol-LLM architecture and training, including the hyperparameters listed in Table 7.

Architecture When using Q-Former as the cross-modal projector, instead of using randomly initialized weights, we initialize it, similarly to Liu et al. [8], using the parameters of a 12-layer pre-trained transformer encoder with an embedding dimension of 768. However, we observed that successful multi-task learning can be achieved without fully utilizing all 12 layers of the Q-Former while maintaining performance without significant performance degradation. Therefore, to reduce the pre-training cost of Q-Former, we use only 5 layers instead of all 12 layers. The number of Q-Former query tokens is set to 32 for multi-task learning, which is more than the eight used in prior work [8]. We set the LoRA rank to 64, alpha to 32, and the dropout rate to 0.1.

Three Stage Training For component ablation, we maintain identical hyperparameters for Mol-LLM, Mol-LLM (w/o Graph), and Mol-LLM (w/o MolPO), as specified in Table 7. In Stage 1, along with the GNN pre-training described in Appendix C.2, we fine-tune only the LoRA parameters of the LLM for 12 epochs. In Stage 2, we train the Q-Former for a single epoch to align the LLM and GNN embeddings learned in Stage 1. Next, in Stage 3, as described in Section 2, we train using the combined objective $\mathcal{L}_{\text{SFT}} + c\mathcal{L}_{\text{MolPO}}$, which combines both the SFT and MolPO objectives. Here, the scaling factor $c = 0.25$ is adjusted to ensure that the scales between $\mathcal{L}_{\text{SFT}}$ and $c\mathcal{L}_{\text{MolPO}}$ do not differ significantly. For the hyperparameters used in $\mathcal{L}_{\text{MolPO}} = \mathbb{E}_{(s,q_i,g,y)\sim\mathcal{D}_{\text{tr}}}[-\log\sigma\big(\min(r_{w,i} - r_{\ell,i}, \lambda_{\text{clip}}|r_{w,i}|) - \gamma_i\big)]$, we use $\lambda_{\text{margin}} = 0.5$ and $\lambda_{\text{clip}} = 1.0$, respectively. For Stage 3, we initially trained the model for 6 epochs using the hyperparameters specified in Table 7; however, we observed

Table 8: Details of Mol-LLM instruction-tuning training data and its sources.

Task	Data Sources	# Train	# Test	# All
Property Prediction (Regression)	MoleculeNet [41]	359,556	2,519	362,075
Property Prediction (Classification)	MoleculeNet [41]	59,607	7,460	67,067
Forward Reaction Prediction	USPTO [34]	1,079,379	5,062	1,084,441
Retrosynthesis	USPTO 500MT	968,943	5,156	974,099
Reagent Prediction	USPTO 500K	121,896	1,000	122,896
Molecule Captioning	ChEBI-20 [53]	58,763	5,793	64,556
Description-Guided Molecule Generation	ChEBI-20	58,763	5,838	64,601
Name Conversion	PubChem [54]	599,767	-	599,767
Overall		3,306,674	40,757	3,347,431

that performance had not fully converged on several tasks. Therefore, we report experimental results based on the model trained for one additional epoch using a reduced initial learning rate of 2×10^{-5} (half of the original value) without a warm-up epoch.

D Molecular Instruction-tuning Dataset

This section describes the construction details of our molecular instruction-tuning dataset, whose statistics are described in Table 8. It covers 21 tasks grouped into eight categories, comprising about 3.3M training and 40K test instances.

D.1 In-distribution Dataset Construction

We integrate molecules for each task from the molecule-oriented datasets Mol-Instructions [8] and SMolInstruct [5]. During this integration process, tasks present in both datasets, such as forward synthesis and molecule captioning, are deduplicated to ensure that molecules included in the test set of one dataset do not appear in the training set of the combined dataset. In this process, we exclude certain tasks that are not directly relevant (e.g., NC-I2F and NC-S2F). For tasks absent in both datasets, such as BACE, molecules are directly extracted from the original data sources to construct the dataset. Finally, we augment the resulting task-specific datasets with instructions using templates adopted and extended from SMolInstruct.

D.2 Out-of-distribution Dataset Construction

LogS - AqSol Dataset To evaluate Mol-LLM on OOD LogS prediction, we use the AqSol dataset [32], which contains multiple water solubility datasets in addition to ESOL. The AqSol dataset is constructed by curating data from 9 different water solubility datasets for 9,982 unique molecules. For our out-of-distribution evaluation on the ESOL dataset, we removed instances from the AqSol dataset that overlap with the ESOL dataset based on the molecule's InChI. Notably, it is common for different prediction datasets to annotate different labels for the same molecule. This occurs due to experimental errors or when LogS labels are predicted based on different prediction models. To ensure high label reliability, we retain 925 molecules whose labels are either unique or have an inter-dataset standard deviation < 0.1.

Reaction Prediction - ORDerly Dataset From Open Reaction Database (ORD) [33], we collected non-USPTO reaction data relevant to forward synthesis and retrosynthesis. Since all reactions in our instruction-tuning dataset are derived from USPTO data, the reactions extracted from non-USPTO sources constitute out-of-distribution (OOD) samples. Then, to ensure no duplication between the collected reaction data and those in Mol-Instructions [7] and SMolInstruct [5], we filtered out reactions from these non-USPTO sources whose input molecule scaffolds overlap with molecules used for reaction prediction training. During this, we first extract data for the forward synthesis task and subsequently ensure that the retrosynthesis reaction data extraction does not duplicate entries already obtained for the forward synthesis. Finally, we apply scaffold splitting to each dataset, resulting in 18K training samples and 5K test samples for forward synthesis, and 54K training samples and 5K test samples for retrosynthesis.

Table 9: Numbers of training and evaluation of impactful molecular tasks, which consist of property classification, property regression, reaction prediction, molecule generation, and molecule captioning, of each model. BioT5+ comprises two separate models, each trained on a distinct group of tasks.

Model	# Train Tasks	# Eval Tasks
BioT5+ [6] (Mol-Instructions)	6	6
BioT5+ (ChEBI-20)	6	6
LlaSMol [5]	10	10
Mol-LLM (Ours)	23	15

Table 10: Summary of baseline models categorized by their input modality and model type.

Model	Input Modality	Task Coverage
InstructMol [9]	1D Sequence & 2D Graph	Specialist
MolCA [8]	1D Sequence & 2D Graph	Specialist
MolT5 [29]	1D Sequence Only	Specialist
MolXPT [43]	1D Sequence Only	Specialist
Mol-Instructions [7]	1D Sequence Only	Semi-Generalist
BioT5+ [6]	1D Sequence Only	Semi-Generalist
GPT-4 (5-shot) [2]	1D Sequence Only	Generalist
Galactica [31]	1D Sequence Only	Generalist
3D-MolM [12]	1D Sequence & 3D Conformer	Generalist
ChemDFM [23]	1D Sequence Only	Generalist
LlaSMol [5]	1D Sequence Only	Generalist
Mol-LLM	1D Sequence & 2D Graph	Generalist

E Experimental Details

This section provides supplementary information necessary for understanding and reproducing the main experiments. In Appendix E.1, we detail the resource requirements and execution times needed to reproduce the main results, followed by Appendix E.2 where we define and categorize the baseline molecular language models based on modality and task coverage. In Appendix E.3, we include full experimental results, whose evaluation metrics are skipped in the main body due to the page limit.

E.1 Resources

All experiments, except for graph encoder pre-training, were conducted on 8 NVIDIA A100 80GB GPUs and an AMD EPYC 7713 64-Core processor with 512GB of RAM. Using this hardware configuration, Stage 1 required 6 days of training, Stage 2 required half a day, and Stage 3 required 12 days to complete. In Stage 1 graph encoder pre-training, GINE training took approximately 18 hours on 4 A100 GPUs, and TokenGT took 19 hours.

E.2 Baseline Models

As described in Section 3.1, we categorize the baseline models into three groups: specialist models, semi-generalist models, and generalist models, based on their level of specialization and task coverage. In addition to the three model categories, we provide a classification based on the type of input modalities. These categorizations are summarized in Table 10.

E.2.1 Categories by Input Modalities

1D Sequence Only Models that rely solely on 1D sequences (e.g., SMILES or SELFIES), which address molecules as strings. This category include Galactica 6.7B [31], GPT-4 [2], Mol-Instructions [7], BioT5+[6], LlaSMol [5], MolT5 [29], MolXPT [43], and ChemDFM [23].

1D Sequence & 2D Graph Models integrate string-based and graph-based representations to capture 2D molecular structure. Representative examples are InstructMol [9], MolCA [8], and GIT-Mol [10]. GIT-Mol additionally exploits molecular images, providing another route to leverage structural information.

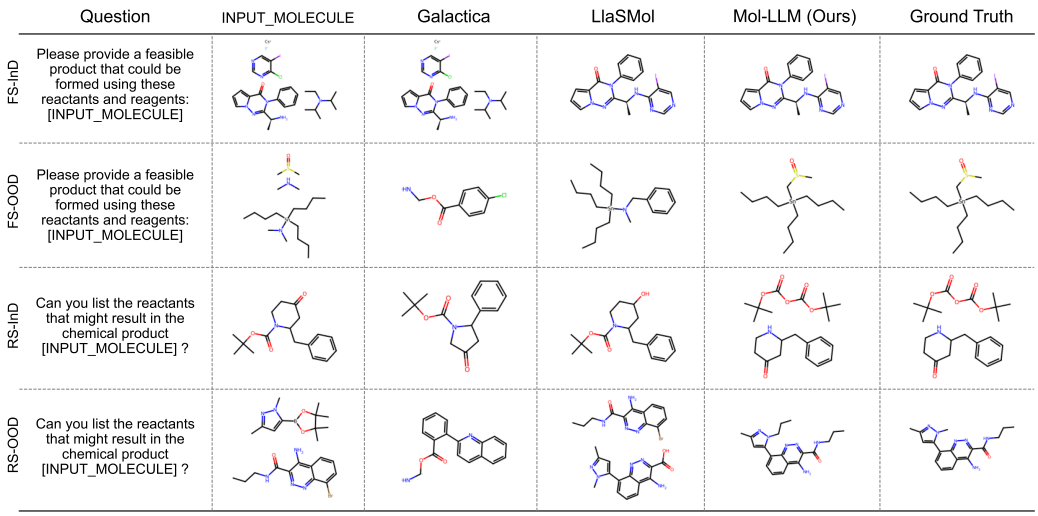

Figure 7: Comparison of predicted outputs by generalists on forward synthesis (FS) and retrosynthesis (RS), both in Mol-Instructions and ORDerly dataset. The upper two rows represent forward synthesis in Mol-Instructions (InD) and ORDerly (OOD) Datasets, respectively, and the lower two rows represent the retrosynthesis task in the same dataset order.

1D Sequence & 3D Conformer Models incorporate 3D conformers alongside sequence information to enrich molecular 3D spatial representations 3D-MoLM [12] belongs to this category.

E.2.2 Categories by Task Coverage

As described in Section 3.1, the baseline models are categorized as follows:

Specialist Models MolCA [8], InstructMol [9], MolXPT [43], GIT-Mol [10], and MolT5 [29] are optimized for individual molecular tasks without parameter or knowledge sharing across tasks.

Semi-Generalist Models BioT5+ [6] and Mol-Instructions [7] address related task groups within a single framework. For instance, BioT5+ trains two separate models: one for classification and translation, and the other for regression and reaction prediction, enabling knowledge sharing within each group while preserving task-specific optimization.

Generalist Models Galactica 6.7B [31], GPT-4 [2], LlaSMol [5], and ChemDFM [23] aim for broad generalization by simultaneously tackling all molecular task groups.

E.3 Full Experimental Results

Table 11 presents the complete results corresponding to Table 2. Table 13 and Table 12 show the full results for Table 5. In Figure 7, we also visualize predicted outputs by generalists, including Mol-LLM, Galactica [31], and LlaSMol [5] on forward reaction prediction and retrosynthesis on both Mol-Instructions and ORDerly datasets.

F Broader Impacts

We currently anticipate no major negative social impacts from this research; nevertheless, there is a possibility that it could be used to generate molecules harmful to humans or the environment. At present, training is carried out on eight NVIDIA A100 GPUs, but scaling to larger LLMs would require additional GPUs and would therefore increase carbon emissions. On the positive side, Mol-LLM enables researchers performing chemical experiments to predict experimental outcomes in advance.

Table 11: Performance comparison on reaction prediction task on Mol-Instructions [7] and SMolInstruct [5] datasets. FS, RS, RP each represent Forward synthesis, Retrosynthesis, and Reagent prediction.

Task	Dataset	Model	EXACT (↑)	BLEU (↑)	RDK FTS (↑)	MACCS FTS (↑)	MORGAN FTS (↑)	VALIDITY (↑)
FS	Mol-Instructions	*Specialist Models*						
		InstructMol	0.536	0.967	0.776	0.878	0.741	1.00
		MolCA*	0.000	0.321	0.329	0.494	0.253	0.01
		Semi-Generalist Models						
		Mol-Instructions*	0.052	0.302	0.232	0.291	0.197	1.00
		BioT5+*(Cls. & Trans.)	0.000	0.206	0.081	0.152	0.069	0.98
		BioT5+*(Reg. & React.)	0.864	0.993	0.949	0.975	0.935	1.00
		Generalist Models						
		GPT-4 (5-shot)	0.021	0.580	0.627	0.728	0.557	0.93
		Galactica	0.000	0.468	0.156	0.257	0.097	0.95
		3D-MoLM*	0.000	0.081	0.223	0.391	0.098	0.01
		ChemDFM*	0.000	0.028	0.104	0.142	0.077	0.07
		LlaSMol*	0.743	0.835	0.920	0.955	0.910	0.95
		Mol-LLM (w/o Graph)	0.893	0.963	0.968	0.983	0.960	**1.00**
		Mol-LLM	**0.911**	**0.969**	**0.976**	**0.987**	**0.967**	**1.00**
	SMolInstruct	*Specialist Models*						
		MolCA*	0.000	0.209	0.252	0.357	0.196	0.01
		Semi-Generalist Models						
		Mol-Instructions*	0.003	0.149	0.139	0.184	0.111	1.00
		BioT5+*(Cls. & Trans.)	0.000	0.286	0.107	0.187	0.089	0.97
		BioT5+*(Reg. & React.)	0.081	0.455	0.418	0.537	0.376	1.00
		Generalist Models						
		GPT-4 (5-shot)	0.011	0.451	0.520	0.634	0.440	0.87
		Galactica*	0.000	0.241	0.292	0.377	0.202	0.36
		3D-MoLM*	0.000	0.086	0.226	0.296	0.117	0.01
		ChemDFM*	0.002	0.046	0.125	0.178	0.109	0.08
		LlaSMol*	**0.629**	**0.883**	**0.871**	**0.919**	**0.848**	0.99
		Mol-LLM (w/o Graph)	0.584	0.867	0.847	0.904	0.815	**1.00**
		Mol-LLM	0.601	0.873	0.853	0.908	0.823	**1.00**
RS	Mol-Instructions	*Specialist Models*						
		InstructMol	0.407	0.941	0.753	0.852	0.714	1.00
		MolCA*	0.000	0.652	0.936	0.880	0.722	0.01
		Semi-Generalist Models						
		Mol-Instructions*	0.069	0.407	0.303	0.359	0.268	1.00
		BioT5+*(Cls. & Trans.)	0.001	0.095	0.114	0.195	0.104	0.97
		BioT5+*(Reg. & React.)	0.642	0.969	0.897	0.930	0.866	1.00
		Generalist Models						
		GPT-4 (5-shot)	0.012	0.573	0.531	0.716	0.506	0.77
		Galactica	0.000	0.452	0.167	0.274	0.134	0.99
		3D-MoLM*	0.000	0.069	0.270	0.451	0.117	0.01
		ChemDFM*	0.000	0.224	0.360	0.440	0.234	0.03
		LlaSMol*	0.453	0.722	0.826	0.885	0.788	0.95
		Mol-LLM (w/o Graph)	0.510	0.839	0.835	0.886	0.797	**1.00**
		Mol-LLM	**0.538**	**0.845**	**0.843**	**0.893**	**0.808**	**1.00**
	SMolInstruct	*Specialist Models*						
		MolCA*	0.000	0.503	0.716	0.760	0.589	0.01
		Semi-Generalist Models						
		Mol-Instructions*	0.015	0.402	0.223	0.285	0.191	1.00
		BioT5+*(Cls. & Trans.)	0.000	0.085	0.095	0.170	0.085	0.97
		BioT5+*(Reg. & React.)	0.152	0.662	0.623	0.751	0.567	1.00
		Generalist Models						
		GPT-4 (5-shot)	0.013	0.523	0.499	0.686	0.465	0.76
		Galactica*	0.000	0.346	0.341	0.447	0.272	0.43
		3D-MoLM*	0.000	0.162	0.220	0.372	0.128	0.01
		ChemDFM*	0.000	0.257	0.304	0.443	0.252	0.03
		LlaSMol*	0.323	0.759	0.749	0.827	0.699	0.99
		Mol-LLM (w/o Graph)	0.363	0.772	0.752	0.828	0.699	**1.00**
		Mol-LLM	**0.377**	**0.779**	**0.760**	**0.832**	**0.707**	**1.00**
RP	Mol-Instructions	*Specialist Models*						
		InstructMol	0.129	0.610	0.444	0.539	0.400	1.00
		MolCA*	0.000	0.002	0.033	0.115	0.012	0.01
		Semi-Generalist Models						
		Mol-Instructions	0.044	0.224	0.237	0.364	0.213	1.00
		BioT5+*(Cls. & Trans.)	0.000	0.169	0.038	0.056	0.015	0.96
		BioT5+*(Reg. & React.)	0.257	0.695	0.539	0.621	0.512	1.00
		Generalist Models						
		GPT-4 (5-shot)	0.000	0.133	0.077	0.228	0.071	0.72
		Galactica	0.000	0.141	0.036	0.127	0.051	0.99
		3D-MoLM*	0.000	0.042	0.039	0.218	0.077	0.01
		ChemDFM*	0.000	0.014	0.033	0.099	0.027	0.06
		LlaSMol*	0.000	0.050	0.041	0.199	0.050	0.93
		Mol-LLM (w/o Graph)	0.202	0.557	0.497	0.586	0.461	**1.00**
		Mol-LLM	**0.225**	**0.578**	**0.517**	**0.600**	**0.485**	**1.00**

Table 12: Performance comparison on description-guided molecule generation task on ChEBI-20 [29] and SMolInstruct [5] datasets.

Dataset	Model	EXACT (↑)	BLEU (↑)	RDK FTS (↑)	MACCS FTS (↑)	MORGAN FTS (↑)	VALIDITY (↑)
ChEBI-20	*Specialist Models*						
	GIT-Mol	0.051	0.756	0.582	0.738	0.519	0.93
	MolT5	0.311	0.854	0.746	0.834	0.684	0.91
	MolXPT	0.215	NA	0.757	0.859	0.667	0.98
	Text+Chem T5	0.322	0.853	0.816	0.901	0.757	0.94
	Semi-Generalist Models						
	Mol-Instructions*	0.016	0.042	0.132	0.167	0.090	1.00
	BioT5+*(Cls. & Trans.)	0.557	0.931	0.835	0.907	0.780	1.00
	BioT5+*(Reg. & React.)	0.537	0.821	0.831	0.897	0.773	1.00
	Generalist Models						
	GPT-4 (5-shot)	0.092	0.485	0.518	0.745	0.482	0.65
	Galactica*	0.000	0.189	0.142	0.264	0.057	0.70
	3D-MoLM*	0.000	0.000	0.000	0.000	0.000	0.00
	ChemDFM*	0.018	0.205	0.136	0.165	0.110	0.19
	LlaSMol*	0.274	0.644	0.755	0.871	0.679	0.95
	Mol-LLM (w/o Graph)	0.431	0.792	0.823	0.903	0.754	**1.00**
	Mol-LLM	**0.443**	**0.795**	**0.829**	**0.906**	**0.761**	**1.00**
SMolInstruct	*Specialist Models*						
	MolT5	0.317	NA	0.802	0.879	0.732	0.95
	Semi-Generalist Models						
	Mol-Instructions*	0.045	0.507	0.366	0.475	0.272	1.00
	BioT5+*(Cls. & Trans.)	0.519	0.918	0.822	0.897	0.757	1.00
	BioT5+*(Reg. & React.)	0.416	0.819	0.782	0.867	0.706	1.00
	Generalist Models						
	GPT-4 (5-shot)	0.027	0.404	0.482	0.726	0.368	0.74
	Galactica*	0.000	0.173	0.144	0.271	0.055	0.61
	3D-MoLM*	0.000	0.000	0.000	0.000	0.000	0.00
	ChemDFM*	0.041	0.069	0.230	0.297	0.189	0.13
	LlaSMol*	0.180	0.718	0.712	0.845	0.623	0.93
	Mol-LLM (w/o Graph)	0.362	0.759	0.797	**0.888**	0.716	**1.00**
	Mol-LLM	**0.368**	**0.761**	**0.800**	0.887	**0.721**	0.99

Table 13: Performance comparison on molecule captioning task on ChEBI-20 [29] and SMolInstruct [5] datasets.

	Model	BLEU-2 (↑)	BLEU-4 (↑)	ROUGE-1 (↑)	ROGUE-2 (↑)	ROUGE-L (↑)	METEOR (↑)
ChEBI-20	*Specialist Models*						
	GIT-Mol	0.352	0.263	0.575	0.485	0.560	0.533
	InstructMol	0.475	0.371	0.566	0.394	0.502	0.509
	MolT5	0.594	0.508	0.654	0.510	0.594	0.614
	MolCA*	0.623	0.540	0.693	0.553	0.631	0.652
	MolXPT	0.594	0.505	0.660	0.511	0.597	0.626
	Text+Chem T5	0.625	0.542	0.682	0.543	0.622	0.648
	Semi-Generalist Models						
	Mol-Instructions	0.249	0.171	0.331	0.206	0.289	0.271
	BioT5+*(Cls. & Trans.)	0.666	0.591	0.709	0.583	0.649	0.680
	BioT5+*(Reg. & React.)	0.249	0.216	0.387	0.302	0.364	0.323
	Generalist Models						
	GPT-4 (5-shot)	0.261	0.158	0.286	0.188	0.303	0.320
	Galactica*	0.001	0.000	0.006	0.000	0.006	0.004
	3D-MoLM*	0.252	0.171	0.361	0.184	0.287	0.326
	ChemDFM*	0.054	0.031	0.120	0.049	0.101	0.078
	LlaSMol*	0.432	0.333	0.522	0.356	0.464	0.466
	Mol-LLM (w/o Graph)	0.556	0.482	**0.565**	**0.417**	**0.509**	0.587
	Mol-LLM	**0.566**	**0.493**	0.493	0.336	0.439	**0.599**
SMolInstruct	*Specialist Models*						
	MolT5	0.462	0.366	0.563	0.398	0.501	0.515
	MolCA*	0.599	0.510	0.665	0.519	0.604	0.628
	Semi-Generalist Models						
	Mol-Instructions	0.028	0.020	0.226	0.160	0.217	0.124
	BioT5+*(Cls. & Trans.)	0.656	0.582	0.702	0.576	0.644	0.677
	BioT5+*(Reg. & React.)	0.257	0.221	0.387	0.301	0.364	0.321
	Generalist Models						
	GPT-4 (5-shot)	0.220	0.125	0.352	0.156	0.273	0.274
	Galactica*	0.002	0.000	0.007	0.000	0.006	0.005
	3D-MoLM*	0.244	0.167	0.357	0.185	0.285	0.329
	ChemDFM*	0.057	0.035	0.128	0.054	0.108	0.085
	LlaSMol*	0.427	0.328	0.525	0.359	0.465	0.470
	Mol-LLM (w/o Graph)	0.554	0.477	**0.544**	**0.393**	**0.490**	0.585
	Mol-LLM	**0.558**	**0.482**	0.485	0.330	0.433	**0.589**