Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multimodal Molecular LLM with Improved Graph Utilization

Sungwoong Kim
swkim01@korea.ac.kr
Korea University
Chanhui Lee
Korea University https://orcid.org/0009-0003-2092-3670
Hanbum Ko
Korea University
Yuheon Song
UNIST https://orcid.org/0009-0003-7281-9285
Yongjun Jeong
Korea University
Rodrigo Hormazabal
LG AI Research
Sehui Han
LG AI Research
Kyunghoon Bae
LG AI Research, Ministry of Science and ICT (MSIT)
Sungbin Lim
LG AI Research, Korea University

Article

Keywords: Molecular LLM, Multimodal LLM, Foundation model, Preference optimization, Generalization

Posted Date: April 28th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8199982/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

Multimodal 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[4,5], Sungbin Lim[4,6*], Sungwoong Kim[1*]

[1*]Department of Artificial Intelligence, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul, Republic of Korea.
[2]Department of Artificial Intelligence, UNIST, 29-1 UNIST-gil, Beomseo-eup, Ulju-gun, Ulsan, Republic of Korea.
[3]Kim Jaechul Graduate School of AI, KAIST, 85 Hoegi-ro, Dongdaemun-gu, Seoul, Republic of Korea.
[4]Materials Intelligence Lab., LG AI Research, 70 Magokjungang 10-ro, Gangseo-gu, Seoul, Republic of Korea.
[5]Ministry of Science and ICT (MSIT), 477 Galmae-ro, Sejong, Republic of Korea.
[6]Department of Statistics, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul, Republic of Korea.

*Corresponding author(s). E-mail(s): sungbin@korea.ac.kr; swkim01@korea.ac.kr;
Contributing authors: chanhui-lee@korea.ac.kr;

Abstract

Large language models (LLMs) are increasingly applied to molecular science, where instruction-tuning enables one model to address diverse tasks, complementing costly or narrow task-specific methods. Yet most molecular LLMs operate on molecular sequences like SMILES, leaving molecular structure reasoning implicit. Recent works augment molecular sequences by concatenating multimodal representations like 2D graphs. However, we observe that naive instruction-tuning through next-token prediction causes models to bypass graph information, including bond connectivity and substructure, which provide important cues for solving diverse molecular tasks. To resolve this, we introduce Molecular structure Preference Optimization (MolPO), a training objective

where an LLM learns to generate answers by conditioning on correct graphs rather than perturbed ones. Furthermore, we introduce a hybrid graph encoder to enhance graph utilization, and leverage extensive instruction-tuning through staged training. The resulting model, Mol-LLM, demonstrates strong performance across broader tasks than prior generalist molecular LLMs, including robust out-of-distribution generalization where they degrade.

Keywords: Molecular LLM, Multimodal LLM, Foundation model, Preference optimization, Generalization

1 Introduction

Large language models (LLMs) [1–4] have been widely used to tackle diverse tasks across domains, such as mathematics and code generation, by leveraging their broad knowledge. This achievement has recently motivated interest in applying LLMs to diverse molecular tasks [5–12] — including chemical reaction prediction, molecular property prediction, description-guided molecule generation, and molecule captioning [5, 7] — all of which are impactful in drug discovery and materials science. However, most molecular LLMs tend to focus on only one of two key components: incorporating *multimodal* molecular information, or performing multitask instruction-tuning as a *generalist* — rather than effectively combining both. Several studies [5, 13] explore the development of generalist molecular LLMs that solve diverse molecular tasks with a single model. They construct instruction-tuning datasets for diverse molecular tasks and fine-tune LLMs for learning transferable and generalizable knowledge. Such generalist molecular LLMs understand and perform various tasks from natural language instructions, substantially outperforming general-purpose LLMs [1–4]. In contrast, others [8–12, 14] extend conventional 1D sequence-based molecular LLMs (e.g., SMILES [15] or SELFIES [16]) to incorporate richer molecular representations. By incorporating 2D molecular graphs or 3D conformers, they represent molecular topology more faithfully [17–19].

However, it is uncertain whether multimodal molecular LLMs effectively utilize molecular structural information when trained with naive supervised fine-tuning (SFT). To investigate this, we introduce the Graph Discrimination Ratio (GDR), which indicates how faithfully a molecular LLM utilizes the molecular graph for a given molecular task. GDR measures how often the model assigns a higher likelihood to the target task answer when conditioned on the original molecular graph, than on its perturbed one. Analysis of GDR on molecular benchmarks (Fig. 1a) shows that the naive SFT model, trained via end-to-end instruction-tuning on downstream tasks as prior works [8, 9, 12, 20, 21], barely distinguishes between original and perturbed graphs across most molecular tasks. More broadly, recent studies [9, 11, 12, 14, 22] have combined multimodal inputs through instruction-tuning, yet whether the resulting LLMs genuinely utilize these inputs has remained unverified.

In this paper, we present Mol-LLM, a multimodal generalist molecular LLM that resolves the graph bypass problem and covers broader molecular tasks than prior

works [5–7, 13]. Specifically, to resolve the graph bypass, we propose a novel Molecular structure Preference Optimization (MolPO), where the LLM is trained to generate target task answers more accurately when conditioned on correct (chosen) molecular graphs than on substructure-perturbed (rejected) ones. In the molecular domain, perturbation must introduce chemically meaningful changes with respect to task-specific properties so that correct and perturbed graphs correspond to distinct answers, providing an effective preference signal. Although molecular substructure perturbation can modify task-specific properties [23], it typically requires molecule-wise analysis tailored to each downstream task. In contrast, MolPO realizes this through a task-agnostic, lightweight MACCS-based perturbation [24] that scales to train molecular foundation LLMs. While MolPO addresses how to learn graph utilization, the quality of the graph representations themselves requires careful encoder design. Built upon a Q-Former-based cross-modal alignment [25], we design a hybrid graph neural network (GNN) that combines GINE [26] for efficient modeling of local structure, and TokenGT [27] for capturing global context, each pre-trained on functional group prediction and molecule reconstruction. On top of these, we utilize an extensive instruction-tuning dataset spanning five task groups: reaction prediction, property regression and classification, description-guided molecule generation, and captioning. To our knowledge, Mol-LLM is among the first multimodal molecular LLMs to combine verified graph utilization across broad molecular benchmarks. Mol-LLM achieves near state-of-the-art performance across diverse molecular tasks, maintaining out-of-distribution (OOD) generalization (Fig. 1b).

In summary, our contributions are:

- **Demonstration of graph bypass.** We empirically demonstrate the graph bypass problem in multimodal molecular LLMs trained with naive instruction-tuning.
- **Resolution of graph bypass.** We resolve the graph bypass problem by proposing Molecular structure Preference Optimization (MolPO), which can scale to foundation model training, combined with a tailored hybrid graph encoder and multi-objective GNN pre-training.
- **Broad competence and generalization.** Through extensive instruction-tuning on broader molecular tasks than prior works, the proposed Mol-LLM with improved graph utilization achieves consistently strong performances and robust OOD generalization.

2 Results

2.1 Experimental Setup

Baseline Models. Regarding task coverage, molecular LLMs are grouped into three categories: specialist, semi-generalist, and generalist. Specialists are trained for a single molecular task; semi-generalists cover a specific set of tasks within a single model but do not span all task groups; and generalists, such as Mol-LLM, are not restricted to specific tasks. Regarding the input modalities, many baseline models rely on molecular 1D sequences, such as SMILES [15] or SELFIES [16]. In contrast, several works

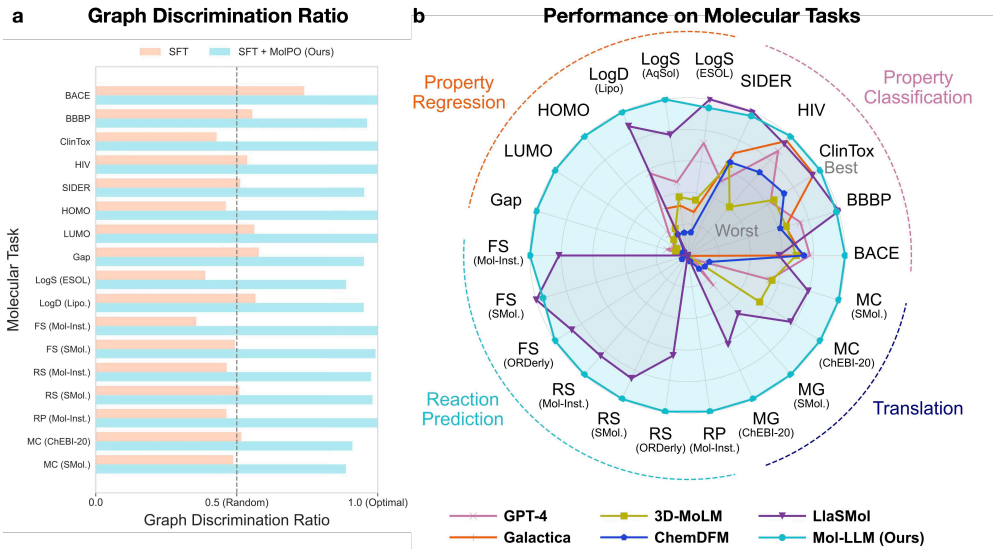


Fig. 1: (a) Graph utilization comparison between SFT and proposed multimodal training (MolPO). A Graph Discrimination Ratio (GDR) near 1 indicates that the model assigns a higher likelihood to the target task answer under the original graph than under the perturbed one, while a GDR near 0.5 indicates random conditioning. (b) Performance comparison among generalist molecular LLMs with normalized primary metrics.

including MolCA [8], InstructMol [9], LLaMo [20], and Mol-LLM, utilize both molecular 1D sequences and 2D graphs. To provide more detailed geometric information than 2D graphs, 3D-MoLM [12] and Mol-LLaMA [21] utilize 3D conformers, involving a trade-off between computational cost and 3D information reliability.

Further details on baseline models including input modalities and task coverage are summarized in Supplementary Table A6 and section A.3.1.

Evaluation Tasks and Datasets. Following prior works [5–8], evaluation tasks and datasets are adopted to provide a comprehensive set over five task groups: reaction prediction, property regression, property classification, description-guided molecule generation, and molecule captioning. Reaction prediction infers a missing reactant, reagent, or product in a chemical reaction equation "{reactant}.{reagent}>>{product}". Forward synthesis (product prediction, FS), retrosynthesis (reactant prediction, RS), and reagent prediction (RP) are all evaluated using datasets from Mol-Instructions [7] and SMolInstruct [5], both of which are constructed from USPTO [28] data. Duplicated instances between train-test splits across datasets are removed to prevent data leakage. Property prediction is divided into two categories: regression of numeric values and binary classification of property presence. The regression benchmark consists of five tasks aligned with previous molecular LLMs: LogS for water solubility (ESOL [5]), LogD for lipophilicity (Lipo [5]),

HOMO [7], LUMO [7], and HOMO-LUMO gap [7]. The classification benchmark comprises BACE [6], BBBP [5], ClinTox [5], HIV [5], and SIDER [5]. Description-guided molecule generation predicts a molecule from a text description; conversely, molecule captioning generates a text description from a molecule. The evaluation benchmarks for molecule generation and captioning comprise ChEBI-20 [29] and SMolInstruct, after removing duplicate molecule instances across the train-test splits of different datasets. Here, PubChem [30], used in some prior works [7, 8], is excluded since its captions are typically short and uninformative for molecular understanding (e.g., "A molecule x can be extracted from x, and the data is available.").

Out-of-distribution Datasets. To assess the out-of-distribution (OOD) robustness, additional evaluation benchmarks are curated for LogS, FS, and RS. For LogS, widely adopted datasets including ESOL are reported to contain inconsistent labels for identical molecules across sources [31, 32]. This limits the reliability of the evaluation based on a single dataset. To this end, our OOD test set is constructed from the AqSol dataset [31], which enables cross-source checks across nine datasets. Molecules with either only one source or label standard deviations above 0.1 across sources are filtered out, retaining 925 molecules from the 9K candidates. To prevent data leakage from ESOL, molecules with InChI [33] overlapping those in ESOL are excluded. For reaction prediction, non-USPTO reactions related to FS and RS are collected from the ORDerly [34]. To ensure structural distributional shifts, reactions whose molecular scaffolds overlap with those in USPTO-derived training data (Mol-Instructions [7] and SMolInstruct [5]) are removed based on scaffold-based filtering. After this scaffold-based filtering, for each of FS and RS, 5K test samples are randomly selected from the remaining 23K and 54K reactions, respectively. This enables OOD evaluation from a structural perspective without data contamination.

Evaluation Metrics and Protocol. For reaction prediction and molecule generation, the following metrics are reported: exact match with the target molecule (EXACT), BLEU [35], and molecular fingerprint similarity using RDKit [36], MACCS [24], and Morgan [37] fingerprints, denoted as RDK FTS, MACCS FTS, and MORGAN FTS, respectively. The proportion of chemically valid molecules (VAL.) is also measured. Due to the space limit, metrics other than EXACT, MACCS FTS, and VAL. are provided in Supplementary Section A.4.5. Similarly, for molecule captioning, BLEU-4, ROUGE-L [38], and METEOR [39] are presented in the main body, while BLEU-2, ROUGE-1, and ROUGE-2 are reported in Supplementary Section A.4.5. For property prediction tasks, following prior works [6, 8, 9], regression performances are evaluated using the root mean squared error (RMSE) or mean absolute error (MAE), while classification performances are measured by the receiver operating characteristic area under the curve (ROC-AUC), computed from the predicted binary probability of the positive class (i.e., "True" token). Especially in classification tasks, ROC-AUC is more comprehensive than binary accuracy used in LlaSMol [5], considering that some datasets (SIDER, ClinTox, and HIV) present significant label imbalance.

Following prior molecular LLM works, our primary results are based on single-seed inference. For property prediction tasks, additional multi-seed results, that are

more aligned with the MoleculeNet evaluation protocol [40, 41], are provided in Supplementary Table A8.

2.2 Graph Utilization Diagnosis and Perturbation Validation

This section demonstrates that the graph bypass problem is resolved via MolPO and that the underlying perturbation strategy is chemically meaningful. Further illustrative examples across diverse tasks are provided in Fig. 4b.

Graph Utilization Diagnosis. To assess whether multimodal molecular LLMs genuinely utilize graph structural information, we propose a likelihood-based measure of graph utilization, Graph Discrimination Ratio (GDR). Given the i-th task with its instruction q_i, target task answer y, the molecular sequence s, the original and perturbed molecular graphs g_w and g_ℓ, GDR for the i-th task is defined as $\mathrm{GDR}_i = \mathbb{E}_{\mathcal{D}_{\mathrm{tr},i}}\left[\mathbb{I}\left[\log \pi_\theta(y \mid s, g_w, q_i) \geq \log \pi_\theta(y \mid s, g_\ell, q_i)\right]\right]$. Here, $\mathbb{I}$ is the indicator function, $\mathcal{D}_{\mathrm{tr},i}$ is a training subset for the i-th task, and π_θ is a model parameterized by θ. GDR measures how often the model predicts the target task answer with higher likelihood under the original graph than under the perturbed one. A GDR near 1 indicates that the model consistently utilizes the task-relevant information from the correct graph, while a GDR of 0.5 indicates no meaningful graph utilization: although substructure perturbation disrupts the chemical consistency between g_ℓ and y, the model assigns equal likelihood to the target task answer under either graph. GDR is compared across two multimodal molecular LLMs: one trained on $\mathcal{L}_{\mathrm{SFT}} + \mathcal{L}_{\mathrm{MolPO}}$, where $\mathcal{L}_{\mathrm{SFT}}$ and $\mathcal{L}_{\mathrm{MolPO}}$ denote training objectives of naive instruction-tuning and MolPO, respectively, and the other with $\mathcal{L}_{\mathrm{SFT}}$ alone following prior works [8, 9, 12, 20, 21]. As shown in Fig. 1a, the model trained with $\mathcal{L}_{\mathrm{SFT}}$ exhibits GDRs near 0.5, whereas the model trained with $\mathcal{L}_{\mathrm{MolPO}}$ achieves notably improved GDRs across tasks. This confirms that MolPO improves graph utilization.

As a complementary analysis, attention scores on graph tokens during answer generation are also compared between the two models (Supplementary Table A7). The model trained with $\mathcal{L}_{\mathrm{MolPO}}$ shows higher attention on graph tokens in 11 out of 12 graph-input tasks, consistent with the result in Fig. 1a.

Chemical Verification of Perturbation. It is inherently challenging to verify that functional-group perturbations, including our MACCS-based perturbation, indeed modify task-specific molecular properties. Since ground-truth task labels (e.g., property values, reaction outcomes) for perturbed molecules are not provided in datasets, resource-intensive data labeling is required, such as wet-lab experiments. While model-based evaluation could serve as a proxy, its reliability is limited by its evaluation accuracy. In contrast, quantum-mechanical property benchmarks (e.g., HOMO, LUMO, and HOMO–LUMO gap) can provide unique opportunities for this verification, since their ground-truth values can be computed via density functional theory (DFT) [42] simulation. DFT calculates the electronic structure of molecules, directly outputting orbital energies including HOMO and LUMO. Accordingly, to reliably verify that our perturbation alters task-relevant molecular properties, we perform DFT calculations at the same level of theory as the original

QM9 benchmark (B3LYP/6-31G*) [43]. Then we compare three quantities based on the properties of HOMO, LUMO, and HOMO–LUMO gap (Table 5): the original QM9 property-label scale, the differences between DFT-derived properties of original and perturbed molecules, and the differences between the DFT-derived properties of original molecules and Mol-LLM's predicted properties. The property difference by our perturbation lies on the order of 10^{-2} Ha, roughly an order of magnitude above the prediction gap ($\sim 10^{-3}$ Ha) and on par with or below the label scale ($\sim 10^{-1}$ Ha for HOMO and the HOMO–LUMO gap, $\sim 10^{-2}$ Ha for LUMO). This indicates that our MACCS-based perturbation produces hard negatives: preference pairs whose task-relevant properties are shifted on a chemically meaningful scale rather than through extreme or label-preserving variations.

2.3 Performances on Molecular Tasks

This section reports performances of Mol-LLM on reaction prediction, property regression and classification, description-guided molecule generation, and molecule captioning, including OOD evaluation in LogS, FS, and RS.

Reaction Prediction. Reaction prediction requires identifying which functional groups participate in the target chemical reaction, making it particularly sensitive to graph utilization. Reflecting this sensitivity, Mol-LLM shows general performance improvement over its sequence-only variant on FS, RS, and RP (Table 1), demonstrating that multimodal training via MolPO and the hybrid GNN effectively enhances the utilization of molecular graph structure. It also outperforms most generalists on all three. Interestingly, among non-generalists, BioT5+ (Reg. & React.) [6], which shows comparable performance to Mol-LLM on the Mol-Instructions dataset, collapses on the SMolInstruct dataset. This drastic performance drop is attributed to BioT5+ being trained on only a small dataset, limiting robustness over diverse molecular distributions.

Property Regression & Classification. Although MoleculeNet [40] is widely used for molecular evaluation, prior molecular LLMs [5, 6, 8, 9], including generalists, only cover a limited set of tasks (Table 2). For comprehensive assessment across diverse scientific use cases, Mol-LLM expands the task coverage of generalist molecular LLMs, consistently achieving best or competitive scores. This demonstrates that the proposed Mol-LLM, combining MolPO, a hybrid GNN architecture, and large-scale training datasets, generalizes robustly across diverse property prediction tasks. While molecular LLM literature primarily reports single-seed results, prior works evaluating on MoleculeNet [40] generally adopt multi-seed evaluation for statistical reliability. To bridge this gap, our multi-seed evaluation results are reported in Supplementary Table A8. The results indicate that seed-induced variance is small relative to performance differences between models, preserving our interpretations above.

Enhanced graph utilization yields notable gains for structurally complex molecules where functional-group perturbation meaningfully alters the target property, as seen in LogS, LogD, and reaction prediction. This effectiveness rests on the perturbed

substructure granularity. An ablation on LogD and LogS shows that functional-group-level perturbation achieves lower RMSE than atom-level perturbation on both tasks (0.949 vs. 1.312 and 1.333 vs. 1.534, respectively; Supplementary Table A9, with setup in Supplementary Section A.4.4), aligning with the chemistry-intuitive view that substructures govern molecular properties [23]. Conversely, the benefit of explicit graph utilization is limited in two cases. First, for small molecules, the limited bond connectivity and substructure complexity allow implicit reasoning over sequences to suffice for prediction, leaving little room for graph utilization [44, 45]. Second, for some binary classification tasks, multiple functional groups jointly determine the target property, so that perturbing only a subset of them may not decisively change it [23]. This joint correspondence is compounded by the limited training data (∼1K samples), together making it harder to learn graph utilization in binary classification.

Description-guided Molecule Generation & Molecule Captioning. On description-guided molecule generation and molecule captioning (Table 3), the advantage of Mol-LLM over generalists arises from two factors: broader training scale and enhanced graph utilization. First, the sequence-only Mol-LLM largely outperforms LlaSMol, reflecting broader task coverage (15 vs. 10 tasks, Supplementary Table A5) and larger training data (3.3M vs. 2M instances, Supplementary Table A4). Second, graph utilization makes Mol-LLM outperform its sequence-only variant in overall captioning quality on balance, though it trails in recall specifically. In captioning, precision measures how often generated tokens match the reference (BLEU), recall measures how much of the reference the caption covers (ROUGE), and METEOR balances the two as an overall quality score. In these terms, a higher METEOR for Mol-LLM than for its sequence-only variant confirms the net benefit, while the split between BLEU and ROUGE reflects a trade-off between precision and recall. BLEU favors precision and ROUGE favors recall, so Mol-LLM produces more concise captions while its sequence-only variant produces more verbose ones that recover more reference tokens. Overall, Mol-LLM largely outperforms other generalists, achieving markedly higher EXACT and METEOR than the second-best generalist LlaSMol.

In molecule generation, where the input is text only, graph-based and sequence-only Mol-LLM perform nearly identically, as expected, showing that Mol-LLM preserves instruction-following when graph inputs are absent.

It is also noteworthy that multimodal molecular LLMs using 3D conformers (3D-MoLM [12], Mol-LLaMA [21]) largely underperform Mol-LLM, despite 3D conformers often being regarded as more informative than 2D graphs. This indicates that utilization of the 3D modality remains underinvestigated.

Generalization Performance. To assess the robustness of Mol-LLM, it is evaluated on three tasks: LogS, FS, and RS (Table 4). The result demonstrates that Mol-LLM remains robust on all three OOD tasks, where other generalists often substantially degrade under shift. Such OOD robustness of Mol-LLM arises from two factors: (i) enhanced graph utilization and (ii) extensive instruction-tuning across diverse molecular tasks. (i) To isolate graph utilization, Mol-LLM is compared with its sequence-only variant, which differs only in graph input involvement. Across all

three OOD tasks, Mol-LLM outperforms its sequence-only variant. Given that they share the entire training data, the result indicates that graph utilization helps the model learn more generalizable molecular representations. (ii) To verify extensive instruction-tuning, the sequence-only Mol-LLM is compared with LlaSMol, both based on the same Mistral 7B backbone [4] and the same input modality. On LogS, they share the same training data, so the superior performance of the sequence-only Mol-LLM reflects transfer from other molecular tasks. On FS and RS, Mol-LLM additionally benefits from more diverse training data.

It is noteworthy that on LogS, Mol-LLM marginally lags LlaSMol on in-distribution (ESOL) but markedly outperforms on the OOD dataset (AqSol). This trend is more compelling considering evaluation reliability: AqSol aggregates nine existing LogS datasets, from which we construct the OOD test set of 925 molecules with cross-source label standard deviations below 0.1 (Section 2.1), whereas ESOL and other LogS benchmarks are reported to exhibit cross-source inconsistency [31, 32]. The same trend holds in FS, where Mol-LLM is slightly weaker on in-distribution (SMol.) but outperforms LlaSMol under OOD (ORDerly). The case study illustrates this generalization (Fig. 4a), with Mol-LLM generating precise output molecules where other generalist models fail.

On the other hand, the semi-generalist BioT5+, which shows comparable performances to Mol-LLM on in-distribution reaction prediction, suffers a large performance drop under OOD evaluation (Table 1). Such a limited robustness strongly motivates large-scale training and generalist modeling to improve generalization.

Effectiveness of MolPO. When the graph is provided alongside the sequence, Mol-LLM outperforms its SFT-only variant on most evaluated tasks (Table 6), confirming that the graph bypass observed under SFT-only training (Fig. 1a) accounts for a measurable performance gap. Notably, improvements span heterogeneous task groups (property prediction, reaction prediction, and molecule generation), indicating that MolPO encourages the model to extract structural information broadly rather than fitting to narrow task groups.

Graph Encoder Design for Multimodal Molecular LLMs. Prior multimodal molecular LLMs [8, 9] commonly adopt off-the-shelf pre-trained graph encoders such as MoleculeSTM [17] and GraphCL [46], yet whether their representations transfer to multimodal LLMs has not been examined. To investigate this, a graph-only variant of Mol-LLM that receives no molecular sequence is evaluated on downstream tasks (HOMO, FS, RP) across architectures (TokenGT, GINE) and weight initializations (our pretraining via functional group prediction and SELFIES reconstruction, MoleculeSTM, GraphCL, and random). Surprisingly, initialization from GraphCL performs similarly to random initialization, and in the case of MoleculeSTM, even worse (Fig. 3b), indicating that off-the-shelf graph encoders are not necessarily beneficial in multimodal molecular LLMs. In contrast, either GINE [26] or TokenGT [27], when trained with our pretraining strategies, achieves the strongest results. The result is attributed to the complementary coverage of local substructure and global molecular context, combined with pre-training objectives aligned to downstream

molecular tasks. These findings support our hybrid graph encoder architecture and pretraining, and more broadly, motivate systematic exploration of encoder design in multimodal molecular LLMs, rather than off-the-shelf adoption.

3 Discussion

This study reveals that multimodal molecular LLMs trained with naive instruction-tuning often bypass molecular graphs, quantified by our GDR. The proposed Molecular structure Preference Optimization (MolPO) resolves this bypass at the scale of foundation model training (Fig. 1a). Mol-LLM achieves near state-of-the-art performances across diverse molecular tasks and generalizes to out-of-distribution settings, confirming that verified graph utilization provides benefit for downstream molecular applications.

Beyond graph utilization, learning graph representations transferable to downstream tasks in each domain requires dedicated investigation. Fig. 3b shows that the common practice of adopting off-the-shelf molecular encoders [8, 9, 12] may limit the effectiveness of multimodal molecular modeling. Pre-training objectives for standalone graph tasks do not necessarily transfer to multimodal molecular LLMs, where the graph encoder must produce representations useful across diverse downstream molecular tasks. Similar concerns arise in other scientific domains, where pre-trained encoders from standalone self-supervised tasks have not been verified for domain applications.

The gains from graph utilization vary, substantial on molecules of complex structure, but limited on simple molecules or on data-limited classification tasks (Table 2). MolPO is most beneficial when (i) molecules are structurally complex enough that implicit sequence-based reasoning is insufficient, (ii) the target property is sensitive to substructure perturbation, and (iii) sufficient training data to learn the preference signal are available. Beyond current community datasets, this calls for larger-scale, structurally diverse molecular data. In addition, evaluation benchmarks covering structurally complex molecules and substructure-sensitive properties would reveal the benefits of structure-aware approaches more clearly.

Several limitations of this study suggest promising future directions. First, Mol-LLM currently leverages 2D graphs; richer modalities offer promising opportunities. 3D conformation remains underexplored (Fig. 1b), and electronic structure information (e.g., electron density, Hamiltonian matrices), which is essential for advanced chemical queries about molecular reactivity and properties, has yet to be integrated. The modality bypass phenomenon diagnosed here could recur for additional modalities, implying that preference-based alignment, such as MolPO, will remain necessary as the modality set grows. Second, perturbation is performed at the functional-group level as a task-agnostic strategy, suited to foundation model training; for highly specific targets where domain knowledge permits fine-grained perturbation design, task-specific strategies may yield additional benefits. Such task-specific design is not explored in this work. Third, evaluation is limited to single-turn instruction-following; multi-step reasoning and multi-turn interaction are not addressed. Impactful tasks such as discovering novel materials or designing full synthesis pathways may require not only richer

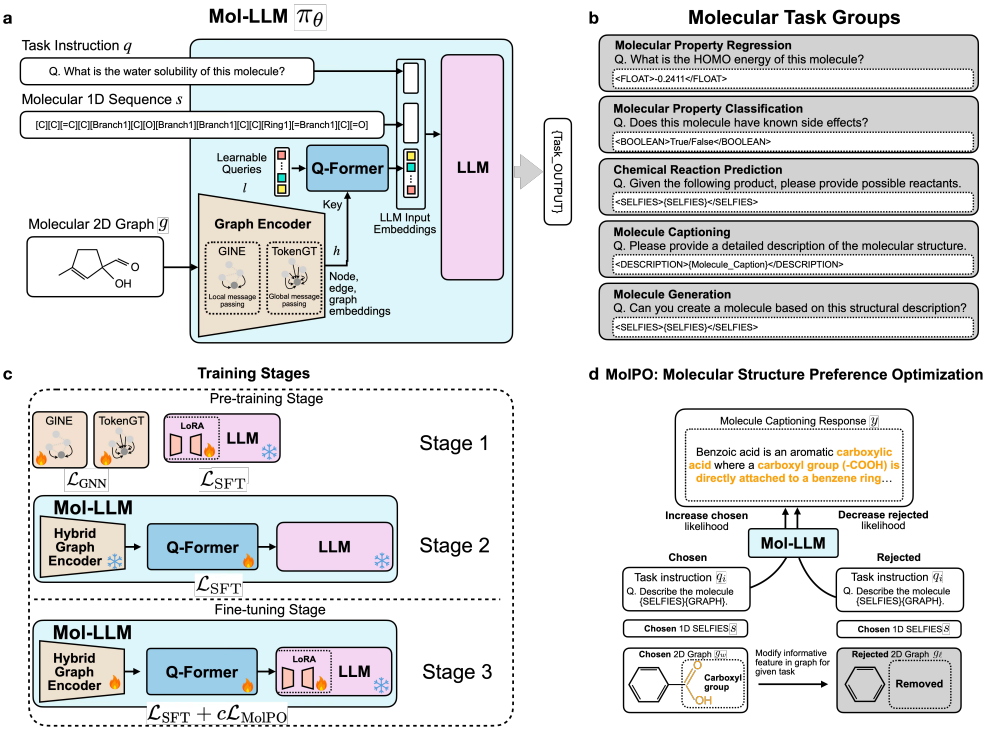


Fig. 2: (a) Overall architecture of Mol-LLM. Molecular graph is encoded into a fixed-length token sequence by a hybrid graph encoder, followed by a Q-Former that outputs query embeddings to feed the LLM, with corresponding task instruction and molecular 1D sequence. (b) Representative downstream molecular tasks. (c) 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. (d) Example of molecular structure preference optimization on a carboxylic acid for the molecular captioning task. Given the carboxylic acid as the chosen graph and the rejected graph where the carboxyl group is removed, LLM learns to decrease the likelihood from the rejected graph while increasing the likelihood from the chosen graph.

modalities but also multi-step reasoning for iterative hypothesis refinement and multi-turn interaction for human-in-the-loop guidance. These capabilities remain largely underdeveloped in molecular LLMs, pointing to promising directions for advancing molecular intelligence.

4 Methods

This section presents Mol-LLM, a multimodal generalist molecular LLM, with its multimodal architecture, training strategy, and instruction-tuning dataset.

4.1 Model Architecture

The multimodal architecture (Fig. 2a) comprises a backbone LLM with a Querying transformer (Q-Former) [25] for cross-modal projection between molecular graphs and text, and a hybrid molecular graph encoder. Within this multimodal architecture, molecular task instructions and 1D molecular sequences are input to the LLM (Fig. 2b), while 2D molecular graphs are embedded by the hybrid graph encoder and aligned to the LLM through the Q-Former. Hyperparameters of model architecture are specified in Supplementary Table A2a.

Backbone Large Language Model. Mol-LLM uses Mistral-7B-Instruct-v0.3 [4] as its backbone LLM, following Yu et al. [5]. To handle molecular tasks efficiently, Mol-LLM extends its tokenizer with the 3K SELFIES vocabulary extracted by BioT5+ [6], and numeric tokens for regression tasks: digits 0–9, a decimal point, and the $+, -$ signs. Mol-LLM further introduces special tokens for SELFIES ("<SELFIES>"), boolean labels ("<BOOLEAN>"), numeric labels ("<FLOAT>"), textual descriptions ("<DESCRIPTION>") and reaction-direction symbol (">>") as depicted in Fig. 2b. These tokens allow it to produce the heterogeneous answer formats required by different downstream tasks.

Hybrid Graph Encoder. Prior multimodal molecular LLMs that use 2D molecular graphs [8, 9] often adopt GINE [26] architecture for efficient modeling of local structure. However, handling diverse tasks and data distributions, including those involving large molecules, also requires a global context. Mol-LLM therefore employs an additional graph encoder, TokenGT [27], a transformer-based graph encoder designed to improve global context understanding and mitigate over-smoothing in large graphs [47], resulting in a hybrid encoder integrating GINE alongside TokenGT. The ablation study on this architectural choice (Fig. 3b) supports the simultaneous use of transformer-based architecture along with GINE. For a 2D molecular graph $g = (v, e)$, GINE f^I outputs a graph embedding $h_g^I \in \mathbb{R}^{1\times d_g}$ and node embeddings $h_v^I \in \mathbb{R}^{|v|\times d_g}$ with embedding dimension d_g, while TokenGT f^T outputs a graph embedding $h_g^T \in \mathbb{R}^{1\times d_g}$ and node embeddings $h_v^T \in \mathbb{R}^{|v|\times d_g}$, and in addition, edge embeddings $h_e^T \in \mathbb{R}^{|e|\times d_g}$. These embeddings $(h_g^I, h_v^I, h_g^T, h_v^T, h_e^T)$ are concatenated along the first dimension to form $h \in \mathbb{R}^{(2|v|+|e|+2)\times d_g}$, which is then used as the key for the Q-Former (Fig. 2a).

Cross-modal Projector. On the cross-modal projector, popular designs include Q-Former adopted in BLIP-2 [25] and the MLP layer adopted in LLaVA [48]. After exploration of the projector types (Supplementary Table A3), Q-Former is adopted due to its superior alignment between graph embedding and target token embedding, as well as its compactness of graph token sequence. Q-Former distills information

from the graph embedding sequence $h \in \mathbb{R}^{(2|v|+|e|+2)\times d_g}$ into the query embedding sequence $l \in \mathbb{R}^{N_q \times d_q}$ via cross-attention, with h as the key and l as the query (Fig. 2a). As a result, the variable length of graph input, scaling with $2|v| + |e|$, is mapped into fixed N_q output tokens. This enables efficient batch processing over training data with heterogeneous molecule sizes. Then, the N_q tokens are concatenated with the SELFIES tokens in various interleaved forms within a given user prompt, and the combined sequence is fed to the LLM. Here, $N_q = 32$ is selected via exploration, since it suffices to capture molecular graph features across diverse tasks.

4.2 Multimodal Training

The proposed multimodal architecture is trained through three stages (Fig. 2c): (Stage 1) Modality-separated pre-training, where the graph encoder and LLM backbone are trained independently, (Stage 2) modality alignment training of Q-Former, between text and graph representations, and (Stage 3) the multimodal training based on MolPO (Fig. 2d). For stages involving LLM training, parameter updates are based on LoRA [49] for training efficiency. Training hyperparameters are specified in Supplementary Table A2b.

Stage 1 - Pre-training for Graph Encoder and LLM Backbone. The pre-training stage trains the LLM backbone and the graph encoder separately (Fig. 2c). The LLM pre-training enables the model to acquire preliminary molecular knowledge efficiently, before the subsequent multimodal training. At this stage, the LLM backbone is pretrained on downstream tasks through next-token prediction. For a training instance with task instruction q, molecular SELFIES string s, and ground truth answer tokens $y = (y_1, \ldots, y_T)$, the instance-wise loss is $\mathcal{L}_{\text{SFT}}^{\text{LLM}} = -\frac{1}{|y|}\sum_{t=1}^{T} \log \pi_\theta^{\text{LLM}}\big(y_t \mid s, q, y_{<t}\big)$ where $\pi_\theta^{\text{LLM}}\big(\cdot \mid s, q, y_{<t}\big)$ denotes the conditional token probability distribution of the LLM backbone parameterized by θ.

Empirically, the graph encoder initialized with pre-trained MoleculeSTM [17] (adopted by prior works MolCA [8], LLaMo [20], and Mol-LLaMA [21]) does not exhibit notable gains in the multimodal molecular LLM, as demonstrated on several targeted molecular tasks (Fig. 3 and Supplementary Table A1). Instead, our graph encoder is pre-trained with two objectives: functional group prediction [50] and SELFIES reconstruction [51], where each training objective is represented as $\mathcal{L}_{\text{func}}$ and $\mathcal{L}_{\text{recon}}$. Functional group prediction enhances expressiveness of functional group representation, often key for many molecular tasks, whereas SELFIES reconstruction encourages the encoder to preserve detailed structural information. Aggregating the two losses, our graph encoder is optimized with $\mathcal{L}_{\text{GNN}} = \mathcal{L}_{\text{func}} + \mathcal{L}_{\text{recon}}$. Ablation study on the graph encoder pre-training (Fig. 3, Supplementary Section A.4.2) and details of the corresponding pre-training dataset and objectives (Supplementary Fig. A1 and section A.2.1) are provided.

Stage 2 - Pre-training for Crossmodal Projector. In Stage 2, only the Q-Former is updated, while the graph encoder and LLM remain frozen. Following the modality alignment in LLaVA [48] rather than the contrastive learning in BLIP-2 [25],

Q-Former is trained via next-token prediction on the downstream task. In our setting, this approach yields better embedding similarity between target text and graph than contrastive learning (Supplementary Table A3). The entire model (combining the graph encoder, Q-Former, and LLM backbone) predicts the conditional token probability π_θ, while only the Q-Former is trained with a token-level cross-entropy loss: $\mathcal{L}_{\text{SFT}} = -\frac{1}{|y|}\sum_{t=1}^{T} \log \pi_\theta\big(y_t \mid s, g, q, y_{<t}\big)$, for one epoch.

Stage 3 - Multimodal Fine-tuning with MolPO. Naive SFT, the multimodal training conducted in prior works [8, 9], suffers from the graph bypass problem: such a molecular LLM fails to exploit the graph modality effectively (Fig. 1a). To resolve the graph bypass, MolPO trains the model to generate the target y by conditioning on the original (chosen) graph $g = g_w$ rather than a perturbed (rejected) graph g_ℓ. Here, for stable and effective preference training, it is crucial that the graph perturbation on g_w produces a g_ℓ that is no longer chemically consistent with the target task answer y. Though modality perturbation has been studied in vision-language by mDPO [52] with random image corruption, such domain-agnostic perturbations cannot reliably break the chemical consistency, leaving the preference signal unreliable for molecular tasks. To this end, given that molecular features are typically attributed to substructures, we construct g_ℓ by perturbing molecular substructures of g_w. Fig. 2d illustrates how the molecular substructure perturbation aligns with preference training: the caption of g_w, describing the carboxyl group, no longer applies to g_ℓ where the carboxyl group is removed. We achieve such graph perturbation via MACCS-based [24] functional-group substitution, which is applicable to any molecular task at minimal computational cost, without the need for task-specific design. Further details on molecular preference pair construction are described in the following Section 4.3.

Upon the g_ℓ construction, the instance-wise rewards for instruction q_i on the i-th molecular task are defined as follows:

$$r_{w,i} = \frac{\beta}{|y|}\sum_t \log \pi_\theta(y_t \mid s, g_w, q_i, y_{<t}), \quad r_{\ell,i} = \frac{\beta}{|y|}\sum_t \log \pi_\theta(y_t \mid s, g_\ell, q_i, y_{<t}), \tag{1}$$

where β is a scaling factor. For efficient large-scale training, we adopt a reference-free reward formulation, as commonly used in methods such as SimPO [53]. Based on the rewards, the instance-wise MolPO loss is defined as

$$\mathcal{L}_{\text{MolPO}} = -\log \sigma\big(\min(r_{w,i} - r_{\ell,i}, \lambda_{\text{clip}}|r_{w,i}|) - \gamma_i\big), \tag{2}$$

with logistic function $\sigma(\cdot)$, clipping coefficient λ_{clip}, and a task-adaptive margin $\gamma_i = \lambda_{\text{margin}}|\mathbb{E}_{(s,g_w,q_i,y)}[r_{w,i}]|$. Given that reward scales differ markedly across tasks (e.g., orders of 10^{-1} and 10^{-2} for regression and reaction prediction, respectively), it is necessary to unify the effective margins across tasks. For this, γ_i is introduced to provide task-adaptive scaling. This is achieved by approximating γ_i with an exponential moving average during training, thereby stabilizing optimization without task-specific hyperparameters. Since lowering $r_{\ell,i}$ is typically easier than increasing $r_{w,i}$, the preference margin $r_{w,i} - r_{\ell,i}$ can be hacked by driving $r_{\ell,i}$ to extreme negatives without a reasonable increase in $r_{w,i}$. This phenomenon was demonstrated by mDPO [52],

which aims to mitigate it via a chosen-reward anchor; however, anchoring $r_{w,i}$ alone leaves $r_{\ell,i}$ unbounded, so the preference margin can still be hacked. To address this directly, MolPO introduces margin clipping in Equation (2), capping the effective gap at $\lambda_{\text{clip}}|r_{w,i}|$. This task-adaptive clipping holds across heterogeneous tasks, so the gap grows only with increasing $r_{w,i}$, not by collapsing $r_{\ell,i}$. Combining MolPO loss with SFT loss, the overall training objective for Stage 3 is $\mathcal{L}_{\text{SFT}} + c\mathcal{L}_{\text{MolPO}}$, where c is a constant. The resulting formulation differs from SimPO [53] and mDPO [52] on three axes: preference focus (graph as sole variation), cross-task scale stabilization (task-adaptive γ_i via EMA), and reward-hacking prevention (margin clipping). A side-by-side comparison is provided in Supplementary Section A.3.4.

Ablations of MolPO loss (Table 6), model architecture and training hyperparameters (Supplementary Table A2) and corresponding detailed descriptions (Supplementary Section A.1.2) are provided.

4.3 Extensive Instruction-tuning Dataset

Molecular Structure Preference Pair. MolPO requires a perturbed molecular graph g_ℓ that is no longer chemically consistent with the target task answer y. By proposing MACCS-based perturbation, we construct g_ℓ in a task-agnostic and low-cost manner. Unlike fragmentation schemes such as BRICS [54] that require per-task analysis and rely on narrow predefined patterns (e.g., only 16 bond-cleavage patterns [23]), the proposed MACCS-based perturbation [24] covers most molecules with 166 structural keys and applies through lightweight RDKit [36] operations. This enables functional-group perturbation at foundation-model scale without per-molecule analysis or task-specific design. Specifically, functional groups identified through MACCS keys are randomly removed and added to the original graph g_w (Fig. 2d), controlled by a functional group sampling ratio τ. Through the perturbed graphs, the model learns to discern structural differences relevant to molecular tasks. For example, MolPO penalizes higher likelihoods when a key functional group, such as a carboxyl group in a captioning task, is removed from the molecule (Fig. 2d). Broader illustrations of our MACCS-based perturbation across diverse tasks are provided in Fig. 4b. We set $\tau = 0.3$, which empirically yields stable convergence of $\mathcal{L}_{\text{MolPO}}$.

Instruction-tuning Dataset. To train Mol-LLM across diverse scientific use cases that encompass the benchmarks of prior multimodal molecular LLMs, an extensive instruction-tuning dataset is constructed. The resulting dataset comprises 15 benchmark tasks and 12 auxiliary tasks for training only, totaling 3.3M training and 40K test instances. We integrate data primarily from Mol-Instructions [7] and SMolInstruct [5]; for mutually overlapping tasks such as forward synthesis and molecule captioning, we deduplicate to prevent any molecule from appearing in both the training and test splits. For tasks not covered by either dataset, such as BACE, data instances are directly sourced following baseline models [6]. In addition, to enhance the understanding of IUPAC [55], we incorporate an IUPAC and SELFIES translation dataset [5]. Each task-specific dataset is then augmented with natural-language instructions extending the templates provided by SMolInstruct. The additional 12 auxiliary tasks include 8 quantum mechanical property regression tasks to support

HOMO, LUMO, and Gap prediction using QM9 labels [43], and 4 translation tasks among molecular sequences to improve their understanding and generation in molecular captioning and description-guided molecule generation. The translation tasks cover IUPAC to SELFIES, IUPAC to molecular formula, SELFIES to IUPAC, and SELFIES to molecular formula, derived from SMolInstruct [5]. Summary statistics are shown in Supplementary Table A4.

5 Related Works

Molecular Large Language Models. MolT5 [29] extends T5 [56] to bidirectional translation between SMILES strings and natural language, whereas MolXPT [57], built on the GPT architecture [58], unifies text–molecule translation with property prediction. MolCA [8] and GIT-Mol [10] fuse 2D molecular graphs with text via a Q-Former [25], while MolLM [59] and Mol-LLaMA [21] further inject 3D geometric cues. InstructMol [9] and PRESTO [60] use a 2-layer MLP as the cross-modal projector, similar to LLaVA [48]. LLaMo [20] introduces an end-to-end multi-level 2D molecular graph projector to reflect hierarchical graph information. UniMoT [11] discretizes Q-Former outputs into graph tokens while 3D-MolT5 [14] introduces 3D structure tokens, enabling generative reasoning over conformers. Although these models utilize molecular structures, each is tailored to a specific set of tasks. In contrast, 3D-MoLM [12] targets broad task coverage with 3D conformers, replacing DFT optimization [42] with faster RDKit-based geometry optimization [36], but at a notable accuracy compromise.

Instruction-tuning on Molecular Tasks. Mol-Instructions [7] introduces 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 1.9M examples across ten tasks, yielding a single model that matches or exceeds specialists. Subsequent works, including UniMoT [11], 3D-MolT5 [14], and 3D-MoLM [12], couple instruction-tuning with 2D/3D structure encoders, yet still lack a systematic strategy for exploiting multimodal inputs, and remain sensitive to shifts in task distribution. 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 across Modalities. In the natural language domain, Direct Preference Optimization [61] aligns LLMs with human preferences by maximizing the log-probability gap between preferred and rejected outputs; SimPO [53] removes the expensive reference model for lighter training. Extending to vision-language, mDPO [52] constructs preference pairs via random image corruption. Applying preference optimization to chemistry, however, requires perturbations that are both scientifically meaningful and scalable across heterogeneous tasks. The proposed MolPO addresses this by adopting MACCS-based perturbation at the functional-group level as a task-agnostic, lightweight strategy that satisfies both requirements.

Data Availability

For reproducibility, all the data required to reproduce the experimental results are available at https://anonymous.4open.science/r/Mol-LLM-submission-01ED, which includes the test dataset and model checkpoints. After acceptance, the train and test datasets, model checkpoints will be released at https://huggingface.co/collections/KU-AGI/mol-llm.

Code Availability

All the code required to understand and reproduce the experimental results is available at https://anonymous.4open.science/r/Mol-LLM-submission-01ED, which covers training, evaluation, data curation, and analysis. After acceptance, the code will be released via a GitHub repository.

Acknowledgements

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.RS2020-II201336, Artificial Intelligence Graduate School Program(UNIST); No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI), 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, 15%)

Author contributions

C.L. conceived the study, developed the methodology, conducted the main experiments, and wrote the manuscript. H.K. contributed to the graph encoder and writing. Y.S. and Y.J. contributed to experiments and writing. R.H., S.H., and K.B. contributed to analysis and writing. S.L. and S.K. directed the research, contributed to analysis and writing, and supervised the project. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Dataset	Mol-Instructions / SMolInstruct					
Task	Forward Synthesis		Retrosynthesis		Reagent Prediction	
Metric	EXACT (↑)	FTS (↑)	EXACT (↑)	FTS (↑)	EXACT (↑)	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						
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						
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
PRESTO*	0.160 / 0.085	0.615 / 0.609	0.318 / 0.127	0.815 / 0.766	0.018	0.373
LLaMo*	0.474 / 0.116	0.882 / 0.700	0.277 / 0.066	0.843 / 0.704	0.000	0.194
Mol-LLaMA*	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	0.000 / 0.000	0.000	0.000
Mol-LLM (w/o Graph)	<u>0.893</u> / 0.584	<u>0.983</u> / 0.904	<u>0.510</u> / <u>0.363</u>	<u>0.886</u> / <u>0.828</u>	<u>0.202</u>	<u>0.586</u>
Mol-LLM	**0.911** / <u>0.601</u>	**0.987** / <u>0.908</u>	**0.538** / **0.377**	**0.893** / **0.832**	**0.225**	**0.600**

Table 1: Performances comparison for reaction prediction tasks on Mol-Instructions [7] and SMolInstruct [5] datasets. Exact match (EXACT) and MACCS fingerprint similarity (FTS) between the predicted molecule and the ground-truth molecule are reported. * denotes results we compute using the official checkpoints; "NA" indicates cases where no official checkpoint is available for evaluation. **Boldface** and <u>underline</u> are the best and the second-best scores among generalist models. For semi-generalist models, each variant is annotated with the task group on which it is trained.

Task	LogS	LogD	HOMO	LUMO	Gap	BACE	BBBP	ClinTox	HIV	SIDER
Metric	RMSE (↓)		MAE (↓)			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										
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										
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**
PRESTO*	≥100	≥100	≥100	≥100	≥100	50.0	50.0	50.0	50.1	50.1
LLaMo*	≥100	3.00	0.0642	0.3680	0.1759	66.0	51.4	65.3	57.4	57.3
Mol-LLaMA*	3.60	4.97	5.3900	9.3900	3.6100	54.2	68.6	49.7	58.2	56.2
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

Table 2: Performances comparison on molecular property prediction tasks from the MoleculeNet [40] benchmark. RMSE (↓) for LogS, LogD, and MAE (↓) for HOMO, LUMO, Gap, and ROC-AUC (↑) for classification tasks from BACE to SIDER. The reported metrics are aligned with prior works for convenient comparisons with prior works. GPT-4 [1] is evaluated with 5-shots, except for classification performances borrowed from Zhao et al. [13] with zero-shot.

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
PRESTO*	0.001 / 0.000	0.314 / 0.304	0.97 / 0.97	0.000 / 0.000	0.007 / 0.007	0.024 / 0.025
LLaMo*	0.000 / 0.000	0.048 / 0.035	0.00 / 0.01	0.133 / 0.146	0.295 / 0.302	0.260 / 0.269
Mol-LLaMA*	0.000 / 0.000	0.000 / 0.000	0.00 / 0.00	0.055 / 0.052	0.204 / 0.201	0.223 / 0.220
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**

Table 3: Performances comparison for molecule generation and molecule captioning on ChEBI-20 [29] and SMolInstruct [5] datasets.

Dataset	AqSol	ORDerly					
Task	LogS	Forward Synthesis			Retrosynthesis		
Metric	RMSE (↓)	EXACT (↑)	FTS (↑)	VAL. (↑)	EXACT (↑)	FTS (↑)	VAL. (↑)
Semi-Generalist							
BioT5+*(Reg. & React.)	1.81	0.095	0.628	1.00	0.139	0.678	1.00
Generalist							
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*	1.39	0.350	0.881	**1.00**	0.473	0.875	0.99
PRESTO*	≥100	0.113	0.669	0.95	0.170	0.670	0.98
LLaMo*	≥100	0.051	0.654	0.60	0.113	0.687	0.21
Mol-LLaMA*	2.41	0.000	0.000	0.00	0.000	0.000	0.00
Mol-LLM (w/o Graph)	1.21	0.394	**0.900**	**1.00**	0.727	0.936	**1.00**
Mol-LLM	**1.17**	**0.401**	0.877	**1.00**	**0.738**	**0.939**	**1.00**

Table 4: Evaluation of Mol-LLM and other baselines for OOD generalization. For reaction prediction, the results on the ORDerly dataset, are reported with exact match (EXACT), MACCS fingerprint similarity (FTS), and validity of predicted molecule (VAL.). For LogS prediction, RMSE on the AqSol dataset is reported.

	HOMO	LUMO	HOMO-LUMO Gap
Label scale (QM9, DFT B3LYP/6-31G)*			
Mean absolute of original label	0.2400	0.0398	0.2512
Perturbation gap (original vs. perturbed, DFT B3LYP/6-31G)*			
MAE between original and perturbed	0.0618	0.0419	0.0754
Prediction gap (original vs. Mol-LLM prediction)			
MAE between original and predicted	0.0044	0.0043	0.0054

Table 5: Chemical verification of functional-group perturbation via DFT calculation on HOMO, LUMO, and HOMO–LUMO gap. Three quantities are compared: (top) the label scale — the mean absolute ground-truth value on QM9 [43]; (middle) the perturbation gap — the MAE in electronic properties between original and functional-group-perturbed molecules; and (bottom) the prediction gap — the MAE between original DFT values and Mol-LLM predictions. The top two rows are computed at the B3LYP/6-31G* level, consistent with the QM9 benchmark. The perturbation gap is on the order of 10^{-2} Ha, roughly an order of magnitude above the prediction gap ($\sim 10^{-3}$ Ha) and on par with or below the label scale ($\sim 10^{-1}$ Ha for HOMO and the HOMO–LUMO gap, $\sim 10^{-2}$ Ha for LUMO). The perturbation gap lies between the prediction gap (below it) and the label scale (at or above it), thereby placing the perturbation-induced shifts within a chemically meaningful range.

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

Table 6: An ablation study on the effect of MolPO on downstream task performances. Root mean squared error (RMSE) (↓) for water solubility (LogS) and lipophilicity (LogD), and exact match (EXACT) (↑) for forward reaction prediction (FS), retrosynthesis (RS), reagent prediction (RP), and description-guided molecule generation (T2M) are reported. "M" and "S" denote the Mol-Instructions and SMolInstruct datasets, respectively.

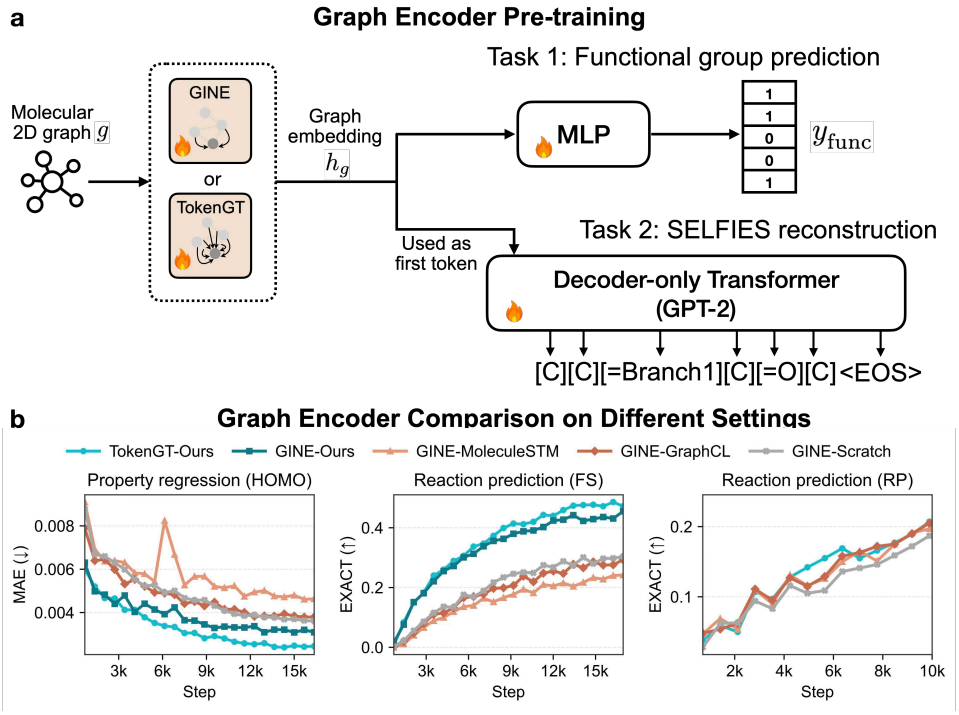


Fig. 3: (a) Overview of the two graph pre-training tasks for the proposed hybrid graph encoder. Two distinct GNN backbones, GINE and TokenGT, are trained independently using two pre-training strategies: functional group prediction and SELFIES reconstruction. (b) Comparison of fine-tuning performances on three tasks under different GNN architectures and initialization with or without pre-trained parameters. *Ours* refers to models initialized with parameters obtained via our GNN pre-training method, whereas *Scratch* denotes models trained from random initialization.

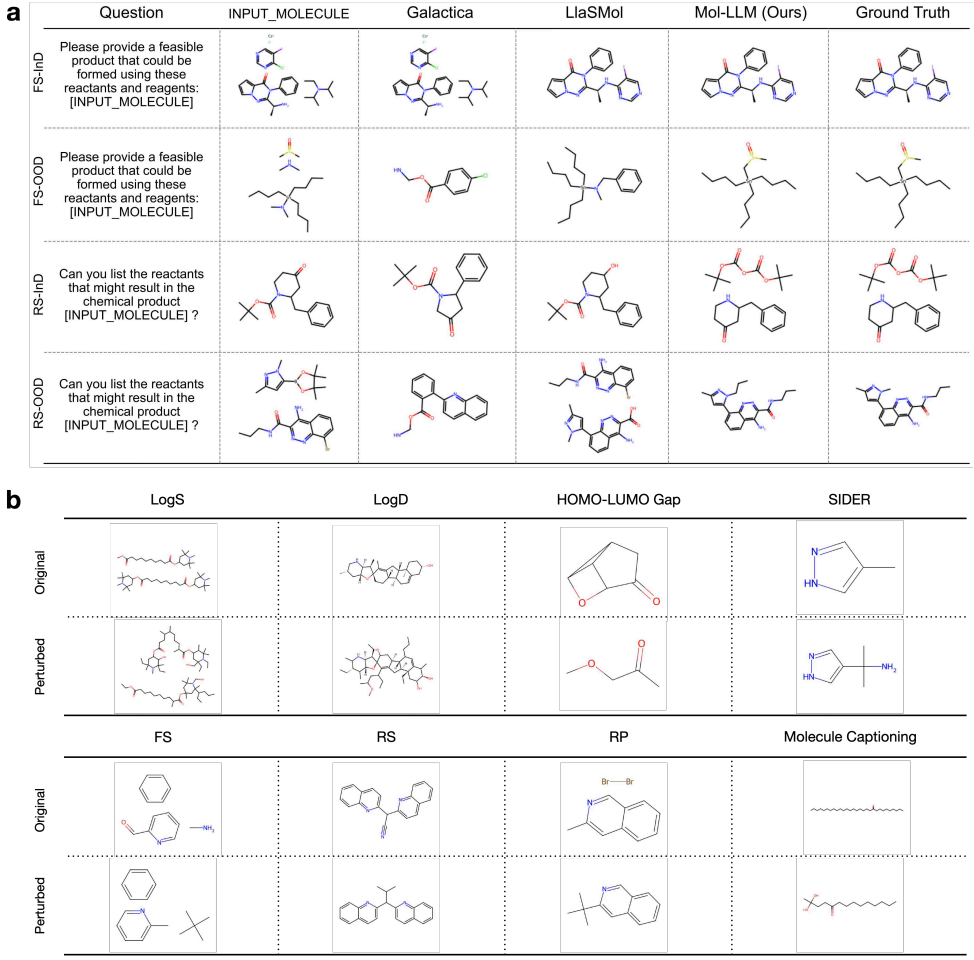


Fig. 4: **(a)** Case study of generalists on InD (Mol-Instructions [7]) and OOD (ORDerly [34]) datasets across forward synthesis (FS) and retrosynthesis (RS). The upper two rows represent forward synthesis in InD and OOD datasets, respectively, and the lower two rows represent the retrosynthesis task in the same order. **(b)** Illustrative examples of functional-group-level perturbation across eight tasks spanning all five task groups: property regression (LogS, LogD, HOMO-LUMO Gap), property classification (SIDER), reaction prediction (FS, RS, RP), and molecule captioning. The original molecule (chosen) and its perturbed counterpart (rejected) are shown, illustrating how functional-group removal or addition alters the chemical context relevant to each task. Examples span a wide range of molecular weights (82–880 Da) and structural complexity — from small heterocycles to large polycyclic and long-chain molecules — to demonstrate that our MACCS-based perturbation produces chemically meaningful preference pairs across diverse molecular scales.

References

[1] OpenAI: Gpt-4 technical report. Preprint at https://arxiv.org/abs/2303.08774 (2023)

[2] Google, G.T.: Gemini: A family of highly capable multimodal models. Preprint at https://arxiv.org/abs/2312.11805 (2023)

[3] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T.: Llama 2: Open Foundation and Fine-Tuned Chat Models (2023). Preprint at https://arxiv.org/abs/2307.09288

[4] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.-A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7B (2023). Preprint at https://arxiv.org/abs/2310.06825

[5] Yu, B., Baker, F.N., Chen, Z., Ning, X., Sun, H.: Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. Preprint at https://arxiv.org/abs/2402.09391 (2024)

[6] Pei, Q., Wu, L., Gao, K., Liang, X., Fang, Y., Zhu, J., Xie, S., Qin, T., Yan, R.: Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. Preprint at https://arxiv.org/abs/2402.17810 (2024)

[7] Fang, Y., Liang, X., Zhang, N., Liu, K., Huang, R., Chen, Z., Fan, X., Chen, H.: Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models. In: ICLR (2024)

[8] Liu, Z., Li, S., Luo, Y., Fei, H., Cao, Y., Kawaguchi, K., Wang, X., Chua, T.-S.: Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. Preprint at https://arxiv.org/abs/2310.12798 (2023)

[9] Cao, H., Liu, Z., Lu, X., Yao, Y., Li, Y.: Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. Preprint at https://arxiv.org/abs/2311.16208 (2023)

[10] Liu, P., Ren, Y., Tao, J., Ren, Z.: Git-mol: A multi-modal large language

model for molecular science with graph, image, and text. Computers in Biology and Medicine **171**, 108073 (2024) https://doi.org/10.1016/j.compbiomed.2024.108073

[11] Zhang, J., Bian, Y., Chen, Y., Yao, Q.: UniMoT: Unified Molecule-Text Language Model with Discrete Token Representation (2024). Preprint at https://arxiv.org/abs/2408.00863

[12] Li, S., Liu, Z., Luo, Y., Wang, X., He, X., Kawaguchi, K., Chua, T.-S., Tian, Q.: 3D-MoLM: Towards 3d molecule-text interpretation in language models. In: ICLR (2024)

[13] Zhao, Z., Ma, D., Chen, L., Sun, L., Li, Z., Xia, Y., Chen, B., Xu, H., Zhu, Z., Zhu, S., et al.: Chemdfm: A large language foundation model for chemistry. Preprint at https://arxiv.org/abs/2401.14818 (2024)

[14] Pei, Q., Wu, L., Gao, K., Zhu, J., Yan, R.: 3D-MolT5: Towards Unified 3D Molecule-Text Modeling with 3D Molecular Tokenization (2024). Preprint at https://arxiv.org/abs/2406.05797

[15] Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. **28**(1), 31–36 (1988) https://doi.org/10.1021/ci00057a005

[16] Krenn, M., Häse, F., Nigam, A., Friederich, P., Aspuru-Guzik, A.: Self-referencing embedded strings (selfies): A 100% robust molecular string representation. Machine Learning: Science and Technology **1**(4), 045024 (2020) https://doi.org/10.1088/2632-2153/aba947

[17] Liu, S., Nie, W., Wang, C., Lu, J., Qiao, Z., Liu, L., Tang, J., Xiao, C., Anandkumar, A.: Multi-modal molecule structure–text model for text-based retrieval and editing. Nat. Mach. Intell. **5**, 1447–1457 (2023) https://doi.org/10.1038/s42256-023-00759-6

[18] Wang, Y., Wang, J., Cao, Z., Farimani, A.B.: Molecular contrastive learning of representations via graph neural networks. Nature Machine Intelligence **4**, 279–287 (2022)

[19] Su, B., Du, D., Yang, Z., Zhou, Y., Li, J., Rao, A., Sun, H., Lu, Z., Wen, J.-R.: A Molecular Multimodal Foundation Model Associating Molecule Graphs with Natural Language (2022). Preprint at https://arxiv.org/abs/2209.05481

[20] Park, J., Bae, M., Ko, D., Kim, H.J.: LLaMo: Large language model-based molecular graph assistant. Advances in Neural Information Processing Systems **37**, (2024)

[21] Kim, D., Lee, W., Hwang, S.J.: Mol-llama: Towards general understanding of

molecules in large molecular language model. Preprint at https://arxiv.org/abs/2502.13449 (2025)

[22] Liang, Y., Zhang, R., Zhang, L., Xie, P.: DrugChat: Towards Enabling ChatGPT-Like Capabilities on Drug Molecule Graphs (2023). Preprint at https://arxiv.org/abs/2309.03907

[23] Wu, Z., Wang, J., Du, H., Jiang, D., Kang, Y., Li, D., Pan, P., Deng, Y., Cao, D., Hsieh, C.-Y., Hou, T.: Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. Nature Communications **14**, 2585 (2023) https://doi.org/10.1038/s41467-023-38192-3

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

[25] Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In: ICML, PMLR 202 (2023)

[26] Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J.: Strategies for pre-training graph neural networks. In: International Conference on Learning Representations (2020)

[27] Kim, J., Nguyen, D., Min, S., Cho, S., Lee, M., Lee, H., Hong, S.: Pure transformers are powerful graph learners. Advances in Neural Information Processing Systems **35**, 14582–14595 (2022)

[28] Wei, J., Yuan, X.-J., Hu, Q., Wang, S.: A novel measure for evaluating classifiers. Expert Syst. Appl. **37**, 3799–3809 (2010)

[29] Edwards, C., Lai, T., Ros, K., Honke, G., Cho, K., Ji, H.: Translation between Molecules and Natural Language. In: Proceedings of EMNLP, pp. 375–413 (2022) https://doi.org/10.18653/v1/2022.emnlp-main.26

[30] Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B.A., Thiessen, P.A., Yu, B., Zaslavsky, L.Y., Zhang, J., Bolton, E.E.: Pubchem 2023 update. Nucleic acids research (2022)

[31] Sorkun, M.C., Khetan, A., Er, S.: Aqsoldb, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. Scientific Data **6** (2019)

[32] Meng, J., Chen, P., Wahib, M., Yang, M., Zheng, L., Wei, Y., Feng, S., Liu, W.: Boosting the predictive performance with aqueous solubility dataset curation. Scientific Data **9**, 71 (2022)

[33] Heller, S.R., McNaught, A., Stein, S., Tchekhovskoi, D.V., Pletnev, I.V.: InChI – the worldwide chemical structure identifier standard. Journal of Cheminformatics **5**, 7–7 (2013). https://api.semanticscholar.org/CorpusID:10769253

[34] Wigh, D.S., Arrowsmith, J., Pomberger, A., Felton, K.C., Lapkin, A.A.: Orderly: Data sets and benchmarks for chemical reaction data. Journal of Chemical Information and Modeling **64**, 3790–3798 (2024)

[35] Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

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

[37] Morgan, H.L.: 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)

[38] Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)

[39] Banerjee, S., Lavie, A.: 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, pp. 65–72 (2005)

[40] Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.S.: Moleculenet: a benchmark for molecular machine learning. Chemical Science **9**, 513–530 (2017)

[41] Zhou, G., Gao, Z., Ding, Q., Zheng, H., Xu, H., Wei, Z., Zhang, L., Ke, G.: Uni-Mol: A universal 3D molecular representation learning framework. International Conference on Learning Representations (2023)

[42] Parr, R.G.: Density-functional theory of atoms and molecules. (1989). https://api.semanticscholar.org/CorpusID:178523579

[43] Ramakrishnan, R., Dral, P.O., Rupp, M., von Lilienfeld, O.A.: Quantum chemistry structures and properties of 134 kilo molecules. Scientific Data **1** (2014) https://api.semanticscholar.org/CorpusID:15367821

[44] Fang, X., Liu, L., Lei, J., He, D., Zhang, S., Zhou, J., Wang, F., Wu, H., Wang, H.: Geometry-enhanced molecular representation learning for property prediction. Nature Machine Intelligence **4**, (2021). https://api.semanticscholar.org/CorpusID:235417265

[45] Mendez-Lucio, O., Nicolaou, C.A., Earnshaw, B.A.: MolE: a foundation model

for molecular graphs using disentangled attention. Nature Communications **15** (2024). https://api.semanticscholar.org/CorpusID:274005862

[46] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in neural information processing systems **33**, 5812–5823 (2020)

[47] Li, Q., Han, Z., Wu, X.-M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[48] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: Advances in Neural Information Processing Systems **36**, pp. 34892–34916 (2023)

[49] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., others, : LoRA: Low-rank adaptation of large language models. ICLR **1**, 3–3 (2022)

[50] Li, B., Lin, M., Chen, T., Wang, L.: Fg-bert: a generalized and self-supervised functional group-based molecular representation learning framework for properties prediction. Briefings in bioinformatics **24 6** (2023)

[51] Tu, Z., Coley, C.W.: Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction. Journal of chemical information and modeling (2021)

[52] Wang, F., Zhou, W., Huang, J.Y., Xu, N., Zhang, S., Poon, H., Chen, M.: mdpo: Conditional preference optimization for multimodal large language models. Preprint at https://arxiv.org/abs/2406.11839 (2024)

[53] Meng, Y., Xia, M., Chen, D.: Simpo: Simple preference optimization with a reference-free reward. Preprint at https://arxiv.org/abs/2405.14734 (2024)

[54] Degen, J., Wegscheid-Gerlach, C., Zaliani, A., Rarey, M.: On the art of compiling and using 'drug-like' chemical fragment spaces. ChemMedChem **3**, 1503–1507 (2008)

[55] Favre, H.A., Powell, W.H.: Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013. Royal Society of Chemistry, Cambridge, UK (2013)

[56] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)

[57] Liu, Z., Zhang, W., Xia, Y., Wu, L., Xie, S., Qin, T., Zhang, M., Liu, T.-Y.: Molxpt: Wrapping molecules with text for generative pre-training. In: ACL

(2023)

[58] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.*: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[59] Tang, X., Tran, A., Tan, J., Gerstein, M.B.: Mollm: a unified language model for integrating biomedical text with 2d and 3d molecular representations. Bioinformatics **40**, 357–368 (2024)

[60] Cao, H., Shao, Y., Liu, Z., Liu, Z., Tang, X., Yao, Y., Li, Y.: PRESTO: Progressive Pretraining Enhances Synthetic Chemistry Outcomes. Findings of the Association for Computational Linguistics: EMNLP 2024 2024. https://aclanthology.org/2024.findings-emnlp.597/,

[61] Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems **36**, 53728–53741 (2023)

Property	Description	Tuning-Full	Tuning-Head (Random GNN)	Tuning-Head (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 capacity 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

Table A1: Comparison of MAE (↓) across different GNN training settings based on unified GINE architecture, on the QM9 dataset. "Tuning-Full" simultaneously tunes both the GNN backbone and the regression head, while "Tuning-Head (Random GNN)" only tunes the regression head while keeping a randomly initialized GNN backbone fixed. "Tuning-Head (Frozen MoleculeSTM)" refers to a model that replaces the randomly initialized GNN backbone in the Tuning-Head approach with a pre-trained MoleculeSTM backbone that remains frozen during training.

Appendix A Supplementary Information

A.1 Implementation Details

A.1.1 Graph Encoder and Pre-training

This section introduces detailed information regarding the graph encoder, including its architecture, pre-training strategy, and the corresponding dataset construction.

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 [1] encoder implemented in `FlashAttention-2` [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. MoleculeSTM adopted in the prior work [3], the pre-trained GNN via graph-text contrastive learning based on molecular captions from PubChem [4], turns out to be without a notable advantage in chosen downstream tasks: HOMO, LUMO, and Gap (Supplementary Table A1). Instead, our graph encoder is pre-trained via functional group prediction and SELFIES reconstruction. Both functional group prediction and SELFIES reconstruction tasks share the same graph-level embedding h_g produced by the GNN (Fig. 3a). 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 functional group index. 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: $\mathcal{L}_{\text{recon}} = -\sum_t \log \pi_\theta^{\text{GPT-2}}\big(s_t \mid h_g,\, s_{<t}\big)$. 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 consists of six layers, eight attention heads, and an embedding size of 1024. The graph encoder trains for 50 epochs with a learning rate of 1×10^{-4}, a batch size of 64, and the AdamW optimizer. Details of the pre-training dataset construction are provided in Supplementary Section A.2.1.

A.1.2 Mol-LLM

Parameter	Value
Q-Former	
bert_hidden_dim	768
bert_name	`scibert` [5]
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

(a) Architectural hyperparameters

Training Parameters	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	0.5
λ_{clip}	NA	NA	1.0
Evaluation Parameters			
gen_max_len		256	
num_beams		1	

(b) Training and evaluation hyperparameters

Table A2: (a) Model hyperparameters used for Mol-LLM architecture, evaluation, and training stages. (b) Training hyperparameters for each stage.

This section describes the details of the Mol-LLM architecture and training, including the hyperparameters listed in Supplementary Table A2.

Architecture. When using Q-Former as the cross-modal projector, instead of using randomly initialized weights, we initialize it, similarly to Liu et al. [3], using the

Embedding Similarity (↑)	SIDER		HOMO		Forward		Molecule Captioning	
	Before	After	Before	After	Before	After	Before	After
2-Layer MLP (Next-token prediction)	0.003	0.022	0.003	0.019	-0.003	0.027	0.001	0.019
Q-Former (Next-token prediction)	-0.005	0.032	-0.003	0.021	-0.005	0.025	0.000	0.023
Q-Former (Contrastive learning)	-0.005	0.004	-0.003	0.003	-0.005	-0.001	0.000	0.001

Table A3: Performance comparison of modality alignment before and after training.

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 8 used in prior work [3]. We set the LoRA rank to 64, alpha to 32, and the dropout rate to 0.1.

Ablation on Modality Alignment. For efficient modality alignment, the modality projector type (Q-Former [6] vs. 2-layer MLP [7]) and pre-training strategies (contrastive learning [6] vs. next-token prediction [7]) are determined through an ablation study (Supplementary Table A3), which evaluates how effectively the projector output graph embeddings align with the answer token embeddings of downstream tasks after Stage 2 training. For next-token prediction, one task from each task group in Mol-LLM is selected, resulting in a training dataset of 40K samples composed of 10K samples each from SIDER, HOMO, forward reaction prediction, and molecule captioning, with training performed for 1 epoch. For contrastive learning, which applies only to text-molecule paired datasets, 10K samples from the molecule captioning dataset are used, and training is performed for 4 epochs to maintain an equivalent number of training steps. The training size mismatch between the next-token prediction and contrastive learning reflects the realistic scenario where a text-molecular paired dataset is harder to procure than simply using downstream tasks. The average cosine similarity between the projector output graph embeddings and the target task label embeddings, measured on the test set both before and after training, is reported (Supplementary Table A3).

Analyzing the differences based on training objectives under the same Q-Former, next-token prediction demonstrates superior modality alignment across all tasks compared to contrastive learning. Although contrastive learning shows a slight increase in alignment after training, a significant difference is observed when compared to the improvement in alignment before and after next-token prediction. This indicates that involving an LLM for modality alignment is more effective than performing contrastive learning with only the Q-Former.

When comparing modality projector types under instruction tuning, Q-Former with instruction tuning generally demonstrates better modality alignment than 2-Layer MLP with instruction tuning. This appears to be because Q-Former better learns graph input representations through the attention mechanism of the transformer encoder. However, for forward reaction prediction, 2-Layer MLP shows better

alignment. This is likely due to the nature of forward reaction prediction, which requires preserving detailed molecular information, and the 2-Layer MLP can utilize embeddings for each node, whereas Q-Former compresses information into 32 fixed query embeddings.

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 Supplementary Table A2. In Stage 1, along with the GNN pre-training described in Section A.1.1, 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 4.2, 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 $\mathcal{L}_{\text{MolPO}} = \mathbb{E}_{(s,g,q_i,y)\sim\mathcal{D}_{\text{tr}}}[-\log\sigma\big(\min(r_{w,i} - r_{\ell,i}, \lambda_{\text{clip}}|r_{w,i}|) - \gamma_i\big)]$, 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 Supplementary Table A2; however, we observed 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.

A.1.3 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 requires 6 days of training, Stage 2 requires half a day, and Stage 3 requires 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.

A.2 Dataset Details

A.2.1 Graph Encoder Pre-training Dataset

As explained in Section 4.2, 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[1], we enumerate 87 functional groups and quantify their occurrences across the entire PubChem database, as summarized in the Supplementary Fig. A1a. Fig. A1a illustrates functional group imbalance, for example, `fr_NH0` (tertiary amines) appears in many molecules, 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

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

groups (from `fr_NH0` to `fr_aryl_methyl`) and the rarest group (`fr_prisulfonamd`), retaining 72 functional groups. Fig. A1b 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 (Fig. A1c) is flatter than before, which enables the graph encoder to learn more unbiased chemical knowledge than when trained on the raw PubChem molecule distribution.

A.2.2 Training Data and Tasks

Mol-LLM training covers 27 tasks in total, amounting to 3.3M instances across property prediction, reaction prediction, molecule captioning, and molecule generation. These consist of 15 benchmark tasks (also used for evaluation) and 12 auxiliary training-only tasks. Per-task training and evaluation sample counts are provided in Supplementary Table A4. Supplementary Table A5 contrasts this task coverage against representative generalist baselines.

Task	Data Sources	# Train	# Test	# All
Property Prediction (Regression)	MoleculeNet [10]	359,556	2,519	362,075
Property Prediction (Classification)	MoleculeNet [10]	59,607	7,460	67,067
Forward Reaction Prediction	USPTO [11]	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 [12]	58,763	5,793	64,556
Description-Guided Molecule Generation	ChEBI-20	58,763	5,838	64,601
Name Conversion	PubChem [4]	599,767	-	599,767
Overall		3,306,674	40,757	3,347,431

Table A4: Details of Mol-LLM instruction-tuning datasets with corresponding data sources.

A.3 Relation to Prior Work

A.3.1 Baseline Models

As described in Section 2.1, we categorize the baseline models into three groups: specialist models, semi-generalist models, and generalist models, based on their level

Model	# Train Tasks	# Eval Tasks
BioT5+ [13] (Mol-Instructions)	6	6
BioT5+ (ChEBI-20)	6	6
LlaSMol [14]	10	10
Mol-LLM (Ours)	27	15

Table A5: 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	Task Coverage	Input Modality
InstructMol [15]	Specialist	1D Sequence & 2D Graph
MolCA [3]	Specialist	1D Sequence & 2D Graph
MolT5 [12]	Specialist	1D Sequence Only
MolXPT [16]	Specialist	1D Sequence Only
Mol-Instructions [17]	Semi-Generalist	1D Sequence Only
BioT5+ [13]	Semi-Generalist	1D Sequence Only
GPT-4 (5-shot) [18]	Generalist	1D Sequence Only
Galactica [19]	Generalist	1D Sequence Only
3D-MoLM [20]	Generalist	1D Sequence & 3D Conformer
ChemDFM [21]	Generalist	1D Sequence Only
LlaSMol [14]	Generalist	1D Sequence Only
PRESTO [22]	Generalist	1D Sequence & 2D Graph
LLaMo [23]	Generalist	1D Sequence & 2D Graph
Mol-LLaMA [24]	Generalist	1D Sequence & 2D Graph & 3D Conformer
Mol-LLM	Generalist	1D Sequence & 2D Graph

Table A6: Summary of baseline models categorized by their task coverage and input modality.

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 Supplementary Table A6.

A.3.2 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 includes Galactica 6.7B [19], GPT-4 [18], Mol-Instructions [17], BioT5+ [13], LlaSMol [14], MolT5 [12], MolXPT [16], and ChemDFM [21].

1D Sequence & 2D Graph. Models integrate string-based and graph-based representations to capture 2D molecular structure. Representative examples are InstructMol [15], MolCA [3], and GIT-Mol [25]. GIT-Mol additionally exploits molecular images, providing another route to leverage structural information.

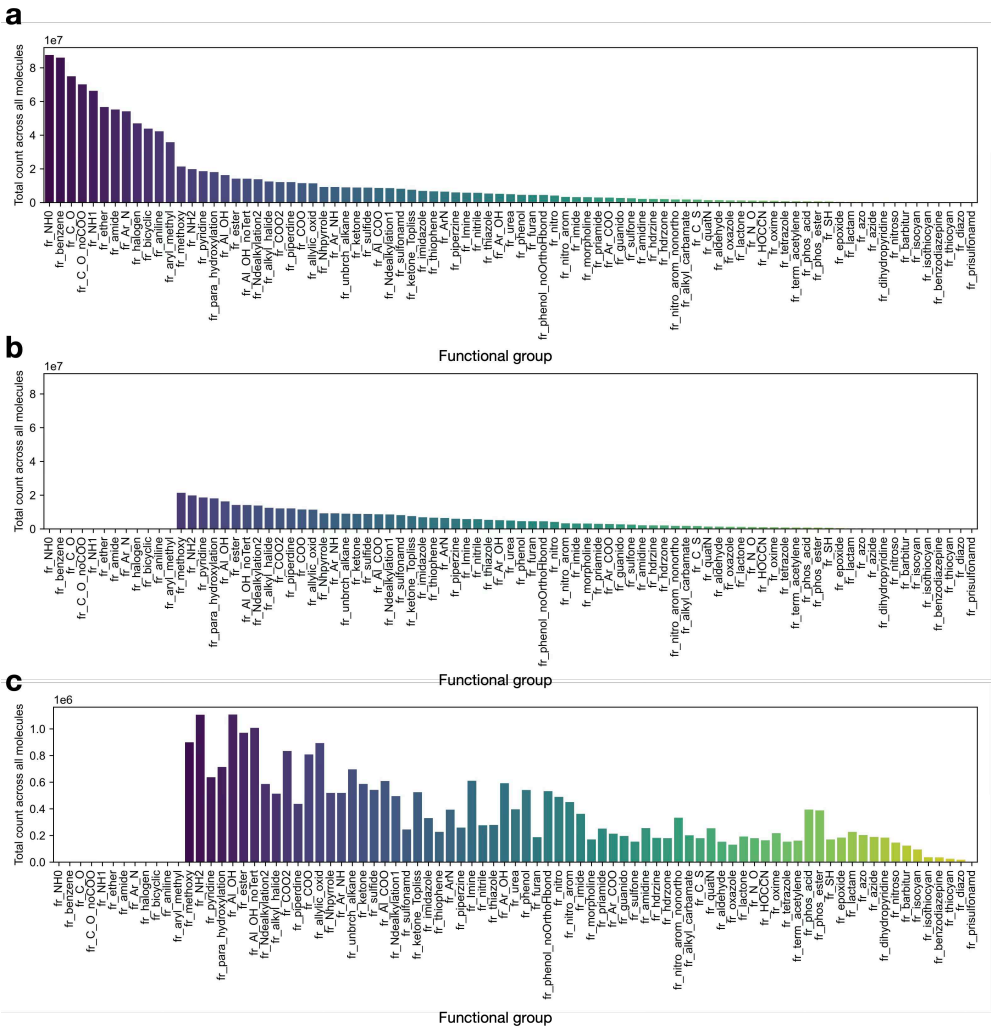


Fig. A1: (a) Distribution of functional groups present in molecules from the PubChem database. (b) Distribution of functional groups in PubChem molecules after excluding groups that are either overly common or extremely rare. (c) Distribution of functional groups obtained after sampling 5M molecules 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.

1D Sequence & 3D Conformer. Models incorporate 3D conformers alongside sequence information to enrich molecular 3D spatial representations. 3D-MoLM [20] belongs to this category.

A.3.3 Categories by Task Coverage

As described in Section 2.1, the baseline models are categorized as follows:

Specialist Models. MolCA [3], InstructMol [15], MolXPT [16], GIT-Mol [25], and MolT5 [12] are optimized for individual molecular tasks without parameter or knowledge sharing across tasks.

Semi-Generalist Models. BioT5+ [13] and Mol-Instructions [17] 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 [19], GPT-4 [18], LlaSMol [14], and ChemDFM [21] aim for broad generalization by simultaneously tackling all molecular task groups.

A.3.4 MolPO Formulation vs. SimPO and mDPO

To clarify how MolPO relates to existing preference-optimization methods, we compare SimPO [8], mDPO [9], and MolPO side-by-side, extending each with the multimodal input of a molecular sequence s and a molecular graph g:

$$
\begin{aligned}
\text{SimPO}: \quad \mathcal{L}_{\text{total}} &= -\log \sigma\big(r(y_w|s,g_w,q_i) - r(y_\ell|s,g_w,q_i) - \gamma_{\text{margin}}\big), \\
\text{mDPO}: \quad \mathcal{L}_{\text{total}} &= -\log \sigma\big(r(y_w|s,g_w,q_i) - r(y_\ell|s,g_w,q_i)\big) \\
&\quad -\log \sigma\big(r(y_w|s,g_w,q_i) - r(y_w|s,g_\ell,q_i)\big) \\
&\quad -\log \sigma\big(r(y_w|s,g_w,q_i) - \delta_{\text{anchor}}\big), \\
\text{MolPO}: \quad \mathcal{L}_{\text{total}} &= -\log \sigma\Big(\min\big(r(y_w|s,g_w,q_i) - r(y_w|s,g_\ell,q_i),\ \lambda_{\text{clip}}|r(y_w|s,g_w,q_i)|\big) - \gamma_i\Big) \\
&\quad -\tfrac{1}{|y_w|}\sum_{t=1}^{|y_w|} \log \pi_\theta\big(y_{w,t} \mid s,g_w,q_i,y_{w,<t}\big),
\end{aligned}
$$

where $r(\cdot) = \frac{\beta}{|y|}\log \pi_\theta(\cdot)$ is the length-normalized log-likelihood for SimPO and MolPO (with $|y|$ denoting the length of the response in the argument), and $r(\cdot) = \beta \log[\pi_\theta(\cdot)/\pi_{\text{ref}}(\cdot)]$ is the DPO-style reward for mDPO. The index i in γ_i denotes the task, and we write $r_{w,i} \equiv r(y_w \mid s, g_w, q_i)$ and $r_{\ell,i} \equiv r(y_w \mid s, g_\ell, q_i)$ for brevity.

The core differences, as summarized in the table below, are: (i) *Preference focus.* MolPO alone varies the input graph (g_w vs. g_ℓ with y_w fixed), focusing on graph bypass; SimPO varies outputs, and mDPO varies both. (ii) *Cross-task reward-scale stabilization.* The task-adaptive γ_i (EMA) unifies reward scales that differ by an order of magnitude (e.g., $\sim 10^{-1}$ for regression vs. $\sim 10^{-2}$ for reaction prediction), preventing severe instability across tasks. SimPO and mDPO instead rely on constant margins

(γ_{margin} and an implicit 0, respectively), which do not adapt to per-task reward magnitudes. (iii) *Reward-hacking prevention.* Reward margin clipping $\min(\cdot, \lambda_{\text{clip}}|r_{w,i}|)$ blocks the optimization from being satisfied by driving $r_{\ell,i}$ to extreme negatives without a corresponding increase in $r_{w,i}$. SimPO does not address this failure mode; mDPO uses a task-agnostic, constant chosen-reward anchor δ_{anchor} that constrains $r_{w,i}$ alone and does not bound $r_{\ell,i}$. This leaves $r_{\ell,i}$ free to collapse faster than $r_{w,i}$ increases. Empirically, removing either γ_i (EMA) or reward margin clipping destabilized training.

	SimPO	**mDPO**	**MolPO**
Preference focus	output: y_w vs. y_ℓ (input g_w fixed)	output: y_w vs. y_ℓ input: g_w vs. g_ℓ	**input only**: g_w vs. g_ℓ (y_w fixed)
Cross-task scale stabilization	constant margin γ_{margin}	constant margin (implicit 0)	**task-adaptive margin** γ_i (EMA)
Reward-hacking prevention	—	constant δ_{anchor} on $r_{w,i}$ only	**task-adaptive reward clipping** ($\lambda_{\text{clip}}|r_{w,i}|$)

A.4 Supporting Analyses and Metrics

This section presents supplementary analyses that support the main findings: attention-based verification of graph utilization, multi-seed robustness evaluation, ablation studies on graph pre-training and perturbation strategy, and complete evaluation metrics.

A.4.1 Attention Score Analysis

A.4.2 Ablation on Graph 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 $\mathcal{L}_{\text{SFT}}$. The resulting performance is then compared with different GNN architectures and weight initializations in Fig. 3b, which 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 [26] and GraphCL [27]) perform either worse than or roughly on par with the non-pretrained baseline *Scratch*, which is a surprising outcome.

Graph Attn. Score	Mol-LLM (w/o MolPO)	Mol-LLM
BACE	0.288	**0.304**
BBBP	0.310	**0.318**
Clintox	0.318	**0.329**
HIV	0.316	**0.332**
SIDER	0.301	**0.312**
HOMO	**0.806**	0.746
LUMO	0.634	**0.666**
Gap	0.488	**0.646**
FS	0.616	**0.679**
RS	0.715	**0.743**
RP	0.853	**0.890**
M2T	0.530	**0.552**

Table A7: Attention-based analysis of graph utilization in Mol-LLM with and without MolPO by calculating attention scores from generated tokens to graph tokens. To determine whether the generated response is actually influenced by graph tokens, attention scores from output tokens to Q-Former embedding tokens are aggregated. The attention scores are averaged across layers, and then, to prevent non-distinctive attention heads from obscuring the overall interpretation, filtered as the top-1 attention score among attention heads. The resulting aggregated attention score is summed across the Q-Former embedding tokens, and reported over 100 test samples per task.

Metric	LogS	LogD	HOMO	LUMO	Gap	BACE	BBBP	ClinTox	HIV	SIDER
	RMSE (↓)			MAE (↓)				ROC-AUC (↑)		
Single inference										
3D-MoLM	3.41	4.86	0.0299	0.0536	0.0673	55.5	53.8	53.7	30.6	49.7
LlaSMol	1.21	1.01	≥1	≥1	≥1	46.7	82.4	77.5	70.3	78.4
PRESTO	≥100	≥100	≥100	≥100	≥100	50.0	50.0	50.0	50.1	50.1
LLaMo	≥100	3.00	0.0642	0.3680	0.1759	66.0	51.4	65.3	57.4	57.3
Mol-LLaMA	3.60	4.97	5.3900	9.3900	3.6100	54.2	68.6	49.7	58.2	56.2
Mol-LLM	1.28	0.91	0.0044	0.0043	0.0054	80.5	81.1	82.4	75.1	76.3
Multi-seed inference (3 seeds, mean ± std)										
Mol-LLM	1.389±.077	0.959±.035	0.0041±.0002	0.0042±.0001	0.0051±.0002	81.2±0.5	82.2±0.8	83.0±0.5	75.8±0.5	75.8±0.3

Table A8: Multi-seed inference evaluation on MoleculeNet property prediction tasks following MoleculeNet evaluation protocol. (Top) Single-inference results of representative baselines and (Bottom) Mol-LLM evaluated with 3 random seeds during inference, reporting mean ± standard deviation. The seed-induced variance is substantially smaller than the performance differences between models, confirming that the comparative interpretations in the main text are robust to seed variation.

A.4.3 Multi-seed Evaluation

The primary evaluations follow the single-seed protocol adopted by prior molecular LLMs [3, 13–15], which has been the standard practice in the molecular LLM literature due to the computational cost of LLM training. However, systematic evaluation of robustness is important for reliable interpretation, particularly for the broader molecular science community where multi-seed evaluation is established practice in domain-specific molecular property prediction studies [10, 28]. Full multi-seed training is computationally prohibitive at the scale of Mol-LLM (3.3M instances, 15 tasks); we

therefore report multi-seed inference evaluation on MoleculeNet property prediction tasks as a best effort toward bridging this gap. This subset is chosen because continuous and threshold-based metrics make seed-induced variance most directly observable, and property prediction on MoleculeNet benchmarks is the setting where multi-seed protocols are most established, enabling systematically reliable evaluation at the level of established domain-specific studies. Aligned with Uni-Mol [28], three different random seeds are applied during LLM generation, and the mean and standard deviation are reported (Supplementary Table A8). The small standard deviations confirm that the main results are robust to seed variation. For example, the seed-induced variance is substantially smaller than the performance differences between Mol-LLM and baseline models, ensuring that the comparative interpretations reported in the main text hold consistently. We provide this analysis to support reproducibility and facilitate independent assessment by readers from both the molecular LLM and the broader molecular modeling communities.

A.4.4 Ablation on Perturbation Strategy

To examine whether the choice of perturbation granularity affects MolPO training, we compare functional-group-level substitution with atom-level substitution under the same 30% replacement ratio (Supplementary Table A9). For atom substitution, alternative atom types are randomly selected from the set of atom types present in the dataset and applied while resampling until no valence bond violations are encountered. Single-atom changes are insufficient to alter molecular properties meaningfully, while substituting a larger fraction of atoms risks disrupting the molecular structure entirely; 30% atom replacement represents a reasonable middle ground. Both strategies are evaluated in a single-task MolPO training setting on LogD and LogS prediction.

Perturbation strategy	LogD (RMSE ↓)	LogS (RMSE ↓)
Functional group substitution (30%)	**0.949**	**1.333**
Atom substitution (30%)	1.312	1.534

Table A9: Comparison of functional-group-level and atom-level perturbation strategies for MolPO training on property prediction tasks. Functional-group perturbation more effectively alters factors directly governing molecular properties (e.g., polarity, hydrogen bonding capability, and ionizability), yielding stronger preference signals than atom-level perturbation.

A.4.5 Full Results

Tables A10 to A12 present the complete results corresponding to Table 1. Table A14 and Table A13 show the full results for Table 3. In Fig. 4a, we also visualize predicted outputs by generalists, including Mol-LLM, Galactica [19], and LlaSMol [14] on forward reaction prediction and retrosynthesis on both Mol-Instructions and ORDerly datasets.

Task		Model	EXACT (↑)	BLEU (↑)	RDK (↑)	MACCS (↑)	MORGAN (↑)	VAL. (↑)
Forward Reaction Prediction	Mol-Instructions	*Specialist*						
		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						
		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						
		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
		PRESTO*	0.160	0.520	0.509	0.615	0.477	0.98
		LLaMo*	0.474	0.805	0.812	0.882	0.789	0.62
		Mol-LLaMA*	0.000	0.000	0.000	0.000	0.000	0.00
		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*						
		MolCA*	0.000	0.209	0.252	0.357	0.196	0.01
		Semi-Generalist						
		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						
		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
		PRESTO*	0.085	0.446	0.501	0.609	0.454	0.96
		LLaMo*	0.116	0.594	0.569	0.700	0.534	0.57
		Mol-LLaMA*	0.000	0.000	0.000	0.000	0.000	0.00
		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**

Table A10: Performance comparison on the forward reaction prediction task using Mol-Instructions [17] and SMolInstruct [14] datasets. RDK, MACCS, and Morgan are different types of molecular fingerprints used to measure molecular similarity.

Task		Model	EXACT (↑)	BLEU (↑)	RDK (↑)	MACCS (↑)	MORGAN (↑)	VAL. (↑)
Retrosynthesis	Mol-Instructions	*Specialist*						
		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						
		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						
		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
		PRESTO*	0.318	0.708	0.720	0.815	0.678	0.97
		LLaMo*	0.277	0.738	0.739	0.843	0.708	0.21
		Mol-LLaMA*	0.000	0.000	0.000	0.000	0.000	0.00
		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*						
		MolCA*	0.000	0.503	0.716	0.760	0.589	0.01
		Semi-Generalist						
		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						
		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
		PRESTO*	0.127	0.584	0.655	0.766	0.600	0.97
		LLaMo*	0.066	0.550	0.568	0.704	0.513	0.19
		Mol-LLaMA*	0.000	0.000	0.000	0.000	0.000	0.00
		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**

Table A11: Performance comparison on retrosynthesis task on Mol-Instructions [17] and SMolInstruct [14] datasets. RDK, MACCS, and Morgan are different types of molecular fingerprint systems used to measure molecular similarity.

Task		Model	EXACT (↑)	BLEU (↑)	RDK (↑)	MACCS (↑)	MORGAN (↑)	VAL. (↑)
Reagent Prediction	Mol-Instructions	*Specialist*						
		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						
		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						
		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
		PRESTO*	0.018	0.253	0.280	0.373	0.274	1.00
		LLaMo*	0.000	0.046	0.041	0.194	0.059	0.27
		Mol-LLaMA*	0.000	0.000	0.000	0.000	0.000	0.00
		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 A12: Performance comparison on the reagent prediction task on Mol-Instructions [17] dataset.

	Model	EXACT (↑)	BLEU (↑)	RDK (↑)	MACCS (↑)	MORGAN (↑)	VAL. (↑)
ChEBI-20	*Specialist*						
	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						
	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						
	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
	PRESTO*	0.001	0.035	0.133	0.314	0.093	0.97
	LLaMo*	0.000	0.000	0.014	0.048	0.006	0.00
	Mol-LLaMA*	0.000	0.000	0.000	0.000	0.000	0.00
	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*						
	MolT5	0.317	NA	0.802	0.879	0.732	0.95
	Semi-Generalist						
	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						
	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
	PRESTO*	0.000	0.028	0.131	0.308	0.089	0.97
	LLaMo*	0.000	0.000	0.011	0.035	0.006	0.01
	Mol-LLaMA*	0.000	0.000	0.000	0.000	0.000	0.00
	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 A13: Performance comparison on description-guided molecule generation task on ChEBI-20 [12] and SMolInstruct [14] datasets.

	Model	BLEU-2 (↑)	BLEU-4 (↑)	ROUGE-1 (↑)	ROUGE-2 (↑)	ROUGE-L (↑)	METEOR (↑)
ChEBI-20	*Specialist*						
	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						
	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						
	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
	PRESTO*	0.003	0.000	0.007	0.000	0.007	0.024
	LLaMo*	0.163	0.133	0.317	0.224	0.295	0.260
	Mol-LLaMA*	0.134	0.055	0.290	0.087	0.204	0.223
	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*						
	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						
	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						
	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
	PRESTO*	0.003	0.000	0.007	0.000	0.007	0.025
	LLaMo*	0.179	0.146	0.325	0.229	0.302	0.269
	Mol-LLaMA*	0.129	0.052	0.285	0.085	0.201	0.220
	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**

Table A14: Performance comparison on molecule captioning task on ChEBI-20 [12] and SMolInstruct [14] datasets.

References

[1] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the North American Chapter of the Association for Computational Linguistics (2019) https://api.semanticscholar.org/CorpusID:52967399

[2] Dao, T.: FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. Preprint at https://arxiv.org/abs/2307.08691 (2023)

[3] Liu, Z., Li, S., Luo, Y., Fei, H., Cao, Y., Kawaguchi, K., Wang, X., Chua, T.-S.: Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. Preprint at https://arxiv.org/abs/2310.12798 (2023)

[4] Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B.A., Thiessen, P.A., Yu, B., Zaslavsky, L.Y., Zhang, J., Bolton, E.E.: Pubchem 2023 update. Nucleic acids research (2022)

[5] Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. In: Conference on Empirical Methods in Natural Language Processing (2019). https://api.semanticscholar.org/CorpusID:202558505

[6] Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In: ICML, PMLR 202 (2023)

[7] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: Advances in Neural Information Processing Systems **36**, pp. 34892–34916 (2023)

[8] Meng, Y., Xia, M., Chen, D.: Simpo: Simple preference optimization with a reference-free reward. Preprint at https://arxiv.org/abs/2405.14734 (2024)

[9] Wang, F., Zhou, W., Huang, J.Y., Xu, N., Zhang, S., Poon, H., Chen, M.: mdpo: Conditional preference optimization for multimodal large language models. Preprint at https://arxiv.org/abs/2406.11839 (2024)

[10] Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.S.: Moleculenet: a benchmark for molecular machine learning. Chemical Science **9**, 513–530 (2017)

[11] Wei, J., Yuan, X.-J., Hu, Q., Wang, S.: A novel measure for evaluating classifiers. Expert Syst. Appl. **37**, 3799–3809 (2010)

[12] Edwards, C., Lai, T., Ros, K., Honke, G., Cho, K., Ji, H.: Translation between Molecules and Natural Language. In: Proceedings of EMNLP, pp. 375–413 (2022) https://doi.org/10.18653/v1/2022.emnlp-main.26

[13] Pei, Q., Wu, L., Gao, K., Liang, X., Fang, Y., Zhu, J., Xie, S., Qin, T., Yan, R.:

Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. Preprint at https://arxiv.org/abs/2402.17810 (2024)

[14] Yu, B., Baker, F.N., Chen, Z., Ning, X., Sun, H.: Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. Preprint at https://arxiv.org/abs/2402.09391 (2024)

[15] Cao, H., Liu, Z., Lu, X., Yao, Y., Li, Y.: Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. Preprint at https://arxiv.org/abs/2311.16208 (2023)

[16] Liu, Z., Zhang, W., Xia, Y., Wu, L., Xie, S., Qin, T., Zhang, M., Liu, T.-Y.: Molxpt: Wrapping molecules with text for generative pre-training. In: ACL (2023)

[17] Fang, Y., Liang, X., Zhang, N., Liu, K., Huang, R., Chen, Z., Fan, X., Chen, H.: Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models. In: ICLR (2024)

[18] OpenAI: Gpt-4 technical report. Preprint at https://arxiv.org/abs/2303.08774 (2023)

[19] Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A.S., Saravia, E., Poulton, A., Kerkez, V., Stojnic, R.: Galactica: A large language model for science. Preprint at https://arxiv.org/abs/2211.09085 (2022)

[20] Li, S., Liu, Z., Luo, Y., Wang, X., He, X., Kawaguchi, K., Chua, T.-S., Tian, Q.: 3D-MoLM: Towards 3d molecule-text interpretation in language models. In: ICLR (2024)

[21] Zhao, Z., Ma, D., Chen, L., Sun, L., Li, Z., Xia, Y., Chen, B., Xu, H., Zhu, Z., Zhu, S., et al.: Chemdfm: A large language foundation model for chemistry. Preprint at https://arxiv.org/abs/2401.14818 (2024)

[22] Cao, H., Shao, Y., Liu, Z., Liu, Z., Tang, X., Yao, Y., Li, Y.: PRESTO: Progressive Pretraining Enhances Synthetic Chemistry Outcomes. Findings of the Association for Computational Linguistics: EMNLP 2024 2024. https://aclanthology.org/2024.findings-emnlp.597/,

[23] Park, J., Bae, M., Ko, D., Kim, H.J.: Llamo: Large language model-based molecular graph assistant. Advances in Neural Information Processing Systems **37**, (2024)

[24] Kim, D., Lee, W., Hwang, S.J.: Mol-llama: Towards general understanding of molecules in large molecular language model. Preprint at https://arxiv.org/abs/2502.13449 (2025)

[25] Liu, P., Ren, Y., Tao, J., Ren, Z.: Git-mol: A multi-modal large language model for molecular science with graph, image, and text. Computers in Biology and Medicine **171**, 108073 (2024) https://doi.org/10.1016/j.compbiomed.2024.108073

[26] Liu, S., Nie, W., Wang, C., Lu, J., Qiao, Z., Liu, L., Tang, J., Xiao, C., Anandkumar, A.: Multi-modal molecule structure–text model for text-based retrieval and editing. Nat. Mach. Intell. **5**, 1447–1457 (2023) https://doi.org/10.1038/s42256-023-00759-6

[27] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in neural information processing systems **33**, 5812–5823 (2020)

[28] Zhou, G., Gao, Z., Ding, Q., Zheng, H., Xu, H., Wei, Z., Zhang, L., Ke, G.: Uni-Mol: A universal 3D molecular representation learning framework. International Conference on Learning Representations (2023)