MolRA: Molecule-Graph Guided Parameter Space Alignment for Molecular Multimodal Large Language Models

Anonymous ACL submission

Abstract

Most existing molecular multimodal large language models rely on input space alignment, where molecular graphs are represented as sequences of continuous embeddings and combined with text prompts into a unified input sequence. This paradigm flattens complex, branched molecular topologies into graph tokens merged with text prompts, requiring the LLM to reconstruct spatial relationships within the shared context window through intensive self-attention, potentially leading to structural fragmentation and increased computational overhead. In this paper, we propose MolRA, a novel molecule-graph guided parameter space alignment approach. Instead of prepending graph tokens, our approach employs a graph-guided weight generator to transform molecular structural features into molecule-graph guided weight signals, which are then injected directly into the decode layers of a frozen LLM. This approach shift decouples molecular graph integration from the input stream and enables molecular structural reasoning without input sequence expansion. Experimental results demonstrate that the proposed approach outperforms recent input-alignment-based molecular multimodal LLMs in both chemical accuracy and instruction-following efficiency. It achieves unified instruction tuning across 11 tasks, attaining state-of-the-art performance on 8 of them, and offers a novel perspective on multimodal integration within scientific domains.

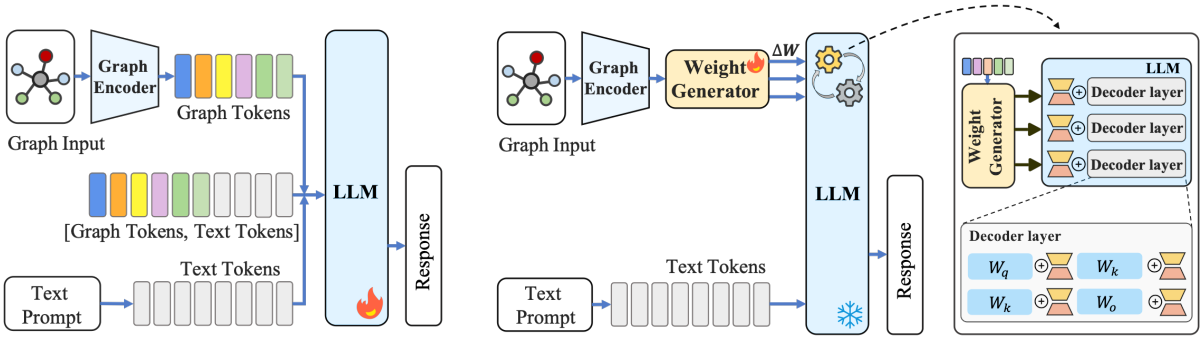
1 Introduction

Multimodal Large Language Models (MLLMs) have profoundly transformed the paradigm of scientific discovery, demonstrating significant breakthroughs across various fields such as medicine, materials science, and biochemistry (Zhou et al., 2024; Pei et al., 2024a; Luo et al., 2023). Adapting Large Language Models (LLMs) for specialized modalities enables general models to perform complex scientific reasoning. To achieve this, establishing precise molecular representation is essential for shifting from text-based modeling toward deep structural reasoning (Liu et al., 2023b, a). This progress has established molecular multimodal modeling as a promising research direction, bridging the modality gap to facilitate tasks such as molecule captioning and text-conditioned compound design (Liu et al., 2023b; Edwards et al., 2022).

To incorporate molecular topological structures, current molecular MLLMs primarily adopt an input space alignment approach, as illustrated in Figure 1a, projecting structural features into the textual input space to construct a unified token sequence. Following the design philosophies of vision-language models, representative studies such as Text+Chem T5 (Edwards et al., 2022), LLaMo (Park et al., 2024), and PRESTO (Park et al., 2024) employ linear projectors or transformer-based adapters to embed graph features into the LLM input stream. This paradigm enables the joint processing of molecular structures and textual instructions within a shared sequence context.

Despite significant advancements in molecular multimodal LLMs, the prevailing input space alignment paradigm introduces inherent computational inefficiencies. The core bottleneck lies in the LLM's attention mechanism, which exhibits quadratic complexity $\mathcal{O}(N^2)$ with respect to sequence length (Vaswani et al., 2017; Tay et al., 2020). For molecular data, linearizing complex, branched graphs into one-dimensional sequences inevitably leads to sequence expansion and structural fragmentation (Edwards et al., 2022). Consequently, the LLM is compelled to reconstruct spatial relationships and multi-hop dependencies through intensive self-attention, significantly increasing computational overhead (Rong et al.,

Code is available at https://anonymous.4open.science/r/MolRA.



(a) Input Space Alignment (b) Parameter Space Alignment

Figure 1: Comparison of multi-modal alignment strategies. **(a)** Baselines align modalities by prepending graph tokens, often weakening textual instructions and structural awareness. **(b)** MolRA injects molecule-aware weight adaptations into a frozen LLM, enabling deep molecular awareness without disturbing the text prompt.

2020; Liu et al., a). As molecular complexity increases, this burden intensifies, severely limiting scalability for large molecules. Even recent methods employing token resampling remain bound by the input alignment paradigm; they still necessitate additional tokens, leaving the fundamental inefficiency unresolved (Pei et al., 2024b). These limitations highlight the critical need for novel strategies that can decouple molecular structural reasoning from input sequence expansion.

To address these challenges, we propose MolRA, a novel framework that shifts molecular graph integration from input space alignment to parameter space alignment (Figure 1b). Instead of introducing additional graph tokens into the input sequence, our approach employs a weight generator to convert molecular structural features into molecule-graph guided weights in the form of low-rank updates (ΔW). This low-rank design ensures computational efficiency when operating within the large parameter space of LLMs. Unlike approaches that rely on fixed projection modules or global adaptation parameters, MolRA dynamically generates updates conditioned on the individual molecular graph, allowing the model to directly modulate its parameters for deep structural reasoning. By directly operating in the parameter space, this approach decouples molecular graph integration from the input sequence, enabling high-fidelity molecular structural reasoning without sequence expansion. Our main contributions are as follows:

- We propose a novel paradigm for molecular multimodal LLMs by shifting molecular graph integration from input space alignment to parameter space alignment.
- Based on this paradigm, we propose MolRA and design an adaptive weight generator that synthesizes molecular structural features into molecule-graph guided weights in the form of low-rank updates.
- Extensive experiments across 11 tasks demonstrate the effectiveness of our approach compared to recent input-alignment-based molecular LLMs.

2 Related Work

2.1 Molecular Large Language Models

Integrating molecular structures with Large Language Models (LLMs) has become a prominent research direction in the pursuit of developing foundational models for science (Cao et al., 2025; Kim et al., 2025; Lee et al., 2025). Prevailing approaches (Edwards et al., 2022; Zeng et al., 2022) have primarily focused on aligning representations from textual descriptions with those from molecular graphs. Early methods, inspired by successes in vision-language pre-training (Dai et al., 2023; Singh et al., 2022), utilized contrastive learning to align global representations of molecular graphs and their textual descriptions (Edwards et al., 2022; Irwin et al., 2022a). However, this global alignment strategy, while effective for retrieval tasks, fails to provide the fine-grained, conditional control necessary for generative tasks such as molecule captioning or reaction prediction (Tran et al., 2025; Gu et al., 2021; Chen et al., 2025). Consequently, the dominant paradigm shifted towards methods that connect a modality-specific encoder to the LLM via a trainable projector module, a strategy popularized by vision-language models like LLaVA (Liu

et al., 2023a). These approaches typically employ a two-stage training process: an initial alignment phase where the projector maps encoded features into the LLM's input space, followed by end-to-end instruction fine-tuning of the entire system. This methodology has become the de facto standard for state-of-the-art multi-modal molecular LLMs (Cao et al., 2025; Le et al., 2024; Lee et al., 2025).

2.2 Parameter-Efficient Fine-Tuning

The high computational and memory costs of large language models (Touvron et al., 2023) render full-parameter fine-tuning impractical for most applications. This challenge has catalyzed the development of Parameter-Efficient Fine-Tuning (PEFT) methods, which adapt pre-trained models by updating only a small subset of their parameters (Han et al., 2024). Prominent PEFT strategies include additive and prefix-based methods. Low-Rank Adaptation (LoRA) (Hu et al., 2022) represents an influential paradigm operating directly in the parameter space. LoRA is predicated on the hypothesis that the weight update matrix during adaptation exhibits a low intrinsic rank. Consequently, it freezes the pre-trained weight matrix $\mathbf{W}_0$ and injects a parallel, trainable low-rank decomposition $\Delta\mathbf{W} = \mathbf{BA}$. During fine-tuning, only the low-rank matrices $\mathbf{A}$ and $\mathbf{B}$ are optimized.

3 Method

3.1 Problem Formulation

We study multi-modal instruction-following for molecular reasoning, where a model generates textual responses conditioned on natural language instructions and molecular structures. Each instance is a triplet $(I, \mathcal{G}, A)$, where I is the instruction, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the molecular graph, and A is the target output. Our goal is to learn a unified model approximating $P(A \mid I; \mathcal{G})$ via the adaptive generator parameters ψ. We maximize the log-likelihood over dataset $\mathcal{D}$:

$$\max_{\psi} \sum_{(I,\mathcal{G},A)\in\mathcal{D}} \log P(A \mid I; \mathcal{G}, \psi). \qquad (1)$$

All base model parameters are frozen, and ψ consists solely of the proposed generator modules, as detailed in Section 3.6.

3.2 MolRA Architecture Overview

MolRA reformulates multi-modal integration by shifting from input-space token prepending to parameter space alignment, thereby eliminating the computational overhead and structural fragmentation caused by sequence expansion. As illustrated in Figure 2, the framework operates through a unified pipeline: The Adaptive Weight Generator (Section 3.3, Fig. 2a) first synthesizes molecular graph embeddings into instance-specific low-rank parameters. These updates are integrated via Dynamic Weight Injection (Section 3.4, Fig. 2b) to modulate the frozen LLM layers. Finally, the Graph-Guided Inference stage (Section 3.5, Fig. 2c) leverages these structure-infused pathways to perform reasoning without altering the textual input sequence.

3.3 Adaptive Weight Generator

The Adaptive Weight Generator is the core of the MolRA, responsible for bridging the modality gap between the graph encoder's feature space and the LLM's parameter space. It transforms the set of graph node embeddings $\mathbf{H}_\mathcal{G}$ into a series of low-rank matrices that modulate the LLM's weights. This process consists of two stages: a cross-attention distillation head and a low-rank parameter projection head.

3.3.1 Graph Encoding

To capture the atomic composition and topological connectivity, we employ a Graph Neural Network (GNN), denoted as $g(\cdot)$, to encode the molecular graph $\mathcal{G}$. The GNN operates via a message-passing scheme over L layers, where the node representations are initialized with atom-level features $\mathbf{h}_v^{(0)}$. At each layer, the representation of every node is iteratively refined by aggregating feature messages from its local neighborhood $\mathcal{N}(v)$ and updating its state. The final set of node embeddings, $\mathbf{H}_\mathcal{G} = \{\mathbf{h}_v^{(L)} \mid v \in \mathcal{V}\}$, effectively encapsulates the molecule's structural topology and serves as the input source for the subsequent Adaptive Weight Generator.

3.3.2 Cross-Attention Distillation Head

Inspired by query-based mechanisms in vision-language models, we distill the node-level graph features into a set of compact representations designed for parameter generation. We introduce a small set of k learnable molecular queries, $\mathbf{Q}_{\text{learn}} \in \mathbb{R}^{k\times d_{\text{model}}}$. These queries are processed by a stack of transformer decoder layers. Within each layer, the queries first attend to each other via self-attention, allowing them to specialize. Subsequently, a cross-attention mechanism enables the queries to attend to the set of node embeddings $\mathbf{H}_\mathcal{G}$, aggregating

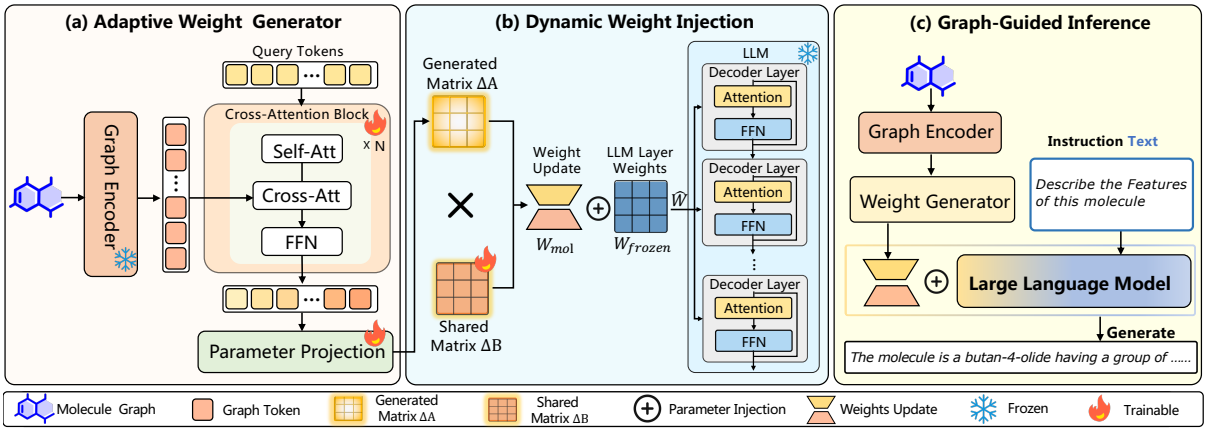


Figure 2: The overall architecture of MolRA. (a) The Adaptive Weight Generator synthesizes structural features into low-rank parameter updates. (b) These adaptive weights are injected into the frozen LLM via parameter space alignment. (c) The inference pipeline for Molecule-Graph Guided reasoning and generation.

relevant structural and chemical information. The output of this process is a set of k context-aware query vectors, $\mathbf{Q}_{\text{out}} \in \mathbb{R}^{k \times d_{\text{model}}}$, which have effectively distilled the molecular structure into a format suitable for the next stage.

3.3.3 Low-Rank Parameter Projection

Generating full-rank weight matrices for the LLM is infeasible. We thus adopt a low-rank factorization strategy inspired by LoRA (Hu et al., 2022). For each target component c in the LLM (e.g., the query projection matrix $\mathbf{W}_q$), the molecule-specific weight update, denoted as $\mathbf{W}_c \in \mathbb{R}^{d_{\text{llm}} \times d_{\text{llm}}}$, is factorized into two smaller matrices, $\Delta\mathbf{A}_c \in \mathbb{R}^{d_{\text{llm}} \times r}$ and $\Delta\mathbf{B}_c \in \mathbb{R}^{r \times d_{\text{llm}}}$, where the rank $r \ll d_{\text{llm}}$. We establish a one-to-one mapping between the k distilled query vectors in $\mathbf{Q}_{\text{out}}$ and the set of all target components. For a given component c, we select its corresponding query vector $\mathbf{q}_c$ from $\mathbf{Q}_{\text{out}}$. The instance-specific matrix $\Delta\mathbf{A}_c$ is generated by projecting $\mathbf{q}_c$ using a fully-connected layer, $\mathbf{W}_{\text{FC}}$, and then reshaping the result. The matrix $\Delta\mathbf{B}_c$ is realized as a shared, learnable linear projector, $\mathbf{W}_{\text{proj}}$, which is constant for all generated weights and is analogous to the up-projection matrix in standard LoRA. The complete weight update for a specific component c, which we denote generically as $\mathbf{W}_{\text{mol}}$, is thus formulated as:

$$\mathbf{W}_{\text{mol}} = \Delta\mathbf{A}_c \cdot \Delta\mathbf{B}_c = R(\mathbf{q}_c\mathbf{W}_{\text{FC}})\mathbf{W}_{\text{proj}}, \quad (2)$$

where $R(\cdot)$ is an operator that reshapes its input vector from $\mathbb{R}^{d_{\text{llm}} \cdot r}$ into a matrix in $\mathbb{R}^{d_{\text{llm}} \times r}$. The layers $\mathbf{W}_{\text{FC}} \in \mathbb{R}^{d_{\text{model}} \times (d_{\text{llm}} \cdot r)}$ and $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{r \times d_{\text{llm}}}$ are learnable parameters. This factorization imposes a strong regularization by decoupling the generation process: the Adaptive Weight Generator only maps the query $\mathbf{q}_c$ into a low-rank bottleneck via $\mathbf{W}_{\text{FC}}$, while the shared $\mathbf{W}_{\text{proj}}$ handles the final projection.

3.4 Dynamic Weight Injection

The molecule-graph guided weight update, $\mathbf{W}_{\text{mol}}$, generated as defined in the preceding section, is dynamically injected into the frozen LLM's parameters. This process creates a temporary, molecule-specific computational path by augmenting the original weights. For a given frozen weight matrix $\mathbf{W}_{\text{frozen}}$ (e.g., $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ in self-attention, and $\mathbf{W}_f$ in MLP projections), the modified weight matrix $\hat{\mathbf{W}}$ used in the forward pass is formulated as a direct summation:

$$\hat{\mathbf{W}} = \mathbf{W}_{\text{frozen}} + \mathbf{W}_{\text{mol}}. \quad (3)$$

This additive modification is applied across designated layers and components of the LLM. By directly absorbing these molecule-aware updates into its parameter space, the LLM can model complex chemical topologies.

3.5 Graph-Guided Inference

As depicted in Figure 2(c), the model executes Graph-Guided Inference using the instruction I as the sole textual input. Unlike input-alignment methods, we avoid sequence expansion by implicitly accessing molecular structures through the modulated parameters $\hat{\mathbf{W}}$. The response A is generated autoregressively:

$$P(a_t \mid a_{<t}, I; \hat{\mathbf{W}}). \quad (4)$$

This design maintains a clean context window, effectively eliminating structural fragmentation and

ensuring the LLM's attention is dedicated entirely to textual reasoning.

3.6 Training Method

The training of MolRA consists of a two-stage process designed to integrate molecular topological structures via parameter space modulation. Throughout both stages, the backbone LLM (θ) and GNN encoder (ϕ) remain frozen, while only the Adaptive Weight Generator (ψ) is updated.

Stage 1: Pre-training for Molecule Text Alignment. This stage bridges the modality gap using molecule text pairs from PubChem (Kim et al., 2023). The Adaptive Weight Generator distills molecular graph topology into molecule-aware parameter modulations, enabling the LLM to reason about chemical properties and bioactivity.

Stage 2: Downstream Tasks Instruction Tuning. We adapt the model to downstream tasks using natural language instructions paired with molecular graphs. This allows MolRA to perform reasoning by leveraging molecule-graph guided weights to modulate the LLM's computational pathways. For each instance $(\mathcal{G}, I, A)$ in the training set $\mathcal{D}$, the model minimizes the standard autoregressive cross entropy loss explicitly with respect to the Adaptive Weight Generator parameters ψ:

$$\mathcal{L}(\psi) = -\sum_{t=1}^{T} \log P(a_t \mid a_{<t}, I; \hat{\mathbf{W}}). \quad (5)$$

This objective realizes effective parameter space alignment by optimizing the transformation of graph structures into weight modulations ($\psi \rightarrow \mathbf{W}_{\text{mol}}$), ensuring high fidelity molecular awareness.

4 Experiments

4.1 Experiment Setup

Benchmarks. We evaluate MolRA across 11 molecular benchmarks grouped into three paradigms to assess comprehensive molecular understanding: (i) Mol2Mol tasks, including forward reaction prediction, retrosynthesis, and reaction condition prediction; (ii) Mol2Text tasks, covering molecule captioning and property description generation; (iii) Mol2Num tasks, focusing on quantitative molecular attribute estimation across diverse datasets. Detailed task statistics and evaluation metrics are provided in Appendix A.1.

Baselines. We compare MolRA with three categories of baselines: (i) previous domain expert models, including MoleculeTransformers (Shin et al., 2019), Chemformer (Irwin et al., 2022b), and Retroformer (Wan et al., 2022); (ii) molecule-specialized LLMs, including TextChemT5 (Christofidellis et al., 2023), T5Chem (Lu and Zhang, 2022), ReactXT (Liu et al., b), MolCA (Liu et al., a), and MoT5-Large (Edwards et al., 2022); and (iii) input-space alignment models, including InstructMol (Cao et al., 2025), PRESTO (Cao et al., 2024), HIGHT (Chen et al., 2024), 3D-MoLM (Li et al.), LLaMo (Park et al., 2024), and UniMoT (Guo et al., 2025).

Evaluation Metrics. Following PRESTO (Cao et al., 2024) and prior work (Fang et al.; Hu et al.), we evaluate MolRA using established task-specific metrics across diverse molecular tasks. For Mol2Mol tasks, we report EXACT, BLEU, Levenshtein distance, fingerprint-based Tanimoto similarities (RDK, MACCS, Morgan), and chemical validity. For Mol2Num tasks, we report the Mean Absolute Error (MAE), except for reaction yield prediction, where we use the coefficient of determination (R^2). For Mol2Text tasks, we adopt standard natural language generation metrics, including BLEU, ROUGE-L, and METEOR. Evaluation metrics details are provided in Appendix A.2.

Implementation Details. Our primary experiments use Vicuna v1.5-7B. In the settings, only the proposed Adaptive Weight Generator is trainable and comprises $N = 8$ decoder blocks, distilling molecular structures into low-rank molecule-aware weight updates with rank $r = 64$. These updates are injected into the $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ and $\mathbf{W}_f$ matrices of the LLM self-attention layers. Following prior work(Cao et al., 2024), we employ SELFIES instead of SMILES for molecular representation to strictly guarantee the chemical validity of generated structures. We optimize the model using AdamW with a learning rate of 2×10^{-5} and a global batch size of 96 across 8 NVIDIA A800 GPUs. Additional implementation details are provided in Appendix A.3.

4.2 Main Results

Tables 1 present the comprehensive evaluation of our framework across diverse molecular understanding and generation tasks. Overall, the proposed parameter space modulation approach consistently achieves superior performance compared to

Table 1: Results on Mol2Mol tasks: forward reaction, retrosynthesis, and condition prediction (reagent, catalyst, solvent). Arrows (↑, ↓) indicate better performance direction. Model denotes domain expert methods; * denotes our re-implementation; and †denotes results from (Cao et al., 2024).

MODEL	EXACT↑	BLEU↑	LEVENSHTEIN↓	RDK FTS↑	MACCS FTS↑	MORGAN FTS↑	VALIDITY↑
Forward Reaction Prediction							
Chemformer† (Irwin et al., 2022b)	0.372	0.824	8.097	0.755	0.820	0.717	0.994
MoleculeTransformers† (Shin et al., 2019)	0.313	0.663	11.735	0.549	0.619	0.532	0.925
LLaMA2†(Touvron et al., 2023)	0.251	0.658	13.167	0.533	0.630	0.512	0.940
Mol-Instruction (Fang et al.)	0.045	0.654	27.262	0.313	0.509	0.262	1.000
nach0-base† (Livne et al., 2024)	0.331	0.857	13.108	0.628	0.709	0.594	0.977
PRESTO† (Cao et al., 2024)	0.355	0.836	10.647	0.646	0.726	0.624	0.973
HIGHT (Chen et al., 2024)	0.293	0.935	16.687	0.774	0.618	0.566	1.000
InstructMol (Cao et al., 2025)	0.536	0.967	10.851	0.776	0.878	0.741	1.000
LLaMo (Park et al., 2024)	0.584	0.894	**6.162**	0.857	0.918	0.841	0.938
UniMoT (Guo et al., 2025)	0.611	0.980	8.297	0.836	0.911	0.807	1.000
MolRA	**0.697**	**0.985**	6.780	**0.875**	**0.923**	**0.843**	**1.000**
Retrosynthesis							
Chemformer†	0.011	0.611	21.073	0.659	0.730	0.574	0.998
Retroformer†(Wan et al., 2022)	0.273	0.769	14.768	0.690	0.782	0.647	0.952
LLaMA2†	0.220	0.754	15.695	0.649	0.747	0.608	0.933
Mol-Instruction	0.039	0.395	31.611	0.279	0.478	0.260	1.000
nach0-base†	0.173	0.854	18.883	0.574	0.668	0.515	0.892
PRESTO†	0.275	0.902	14.433	0.655	0.747	0.619	0.980
HIGHT	0.202	0.914	20.194	0.772	0.623	0.577	0.999
InstructMol	0.407	0.941	13.967	0.753	0.852	0.714	1.000
LLaMo	0.341	0.830	12.263	0.793	0.868	0.750	0.954
UniMoT	0.478	**0.974**	11.634	0.810	0.909	0.771	1.000
MolRA	**0.530**	0.960	**10.720**	**0.820**	**0.918**	**0.782**	**1.000**
Reagent Prediction							
LLaMA2†	0.000	0.283	53.510	0.136	0.294	0.11	1.000
Mol-Instruction	0.044	0.224	23.167	0.237	0.364	0.213	1.000
nach0-base†	0.001	0.172	34.212	0.053	0.134	0.039	0.932
T5Chem	0.019	0.559	11.044	0.366	0.461	0.374	0.994
PRESTO†	**0.275**	**0.902**	**14.433**	**0.655**	**0.747**	**0.619**	0.980
HIGHT	0.007	0.482	27.167	0.462	0.346	0.303	1.000
InstructMol	0.129	0.610	19.664	0.444	0.539	0.400	1.000
LLaMo*	0.223	0.862	15.263	0.593	0.668	0.580	0.954
MolRA	0.175	0.689	16.634	0.530	0.589	0.468	**1.000**
Catalyst Prediction							
LLaMA2†	0.680	0.720	2.545	0.882	0.868	0.687	1.000
Mol-Instruction	0.0	0.110	28.424	0.031	0.045	0.015	0.999
nach0-base†	0.0	0.072	36.442	0.129	0.055	0.009	0.849
T5Chem	0.02	0.346	13.41	0.146	0.268	0.20	0.99
PRESTO†	0.768	0.814	1.755	0.914	0.895	0.774	1.000
LLaMo*	0.737	0.770	1.763	0.913	0.868	0.745	0.995
MolRA	**0.775**	**0.824**	**1.706**	**0.923**	**0.891**	**0.782**	**1.000**
Solvent Prediction							
LLaMA2†	0.311	0.462	3.819	0.452	0.480	0.417	1.000
Mol-Instruction	0.0	0.155	25.117	0.030	0.122	0.035	1.000
nach0-base†	0.0	0.072	36.442	0.129	0.055	0.009	0.849
T5Chem†	0.083	0.311	16.22	0.458	0.424	0.40	0.99
PRESTO†	**0.419**	**0.695**	2.758	0.529	**0.547**	**0.506**	0.912
LLaMo*	0.405	0.690	**2.653**	**0.533**	0.538	0.506	**1.000**
MolRA	0.353	0.662	3.697	0.494	0.517	0.462	1.000

Table 2: Performance comparison on Mol2Text tasks, including description generation, molecular captioning, and experimental procedure tasks.

MODEL	B-4↑	R-1↑	R-2↑	R-L↑	M↑
Description Q&A Task					
LLaMA2	0.232	0.351	0.221	0.304	0.469
3D-MoLM(S)	0.261	0.401	0.256	0.346	0.522
3D-MoLM(G)	0.261	0.401	0.259	0.350	0.519
InstructMol	0.117	0.273	0.118	0.178	0.213
LLaMo*	0.463	0.538	0.481	0.565	**0.602**
UniMoT	0.238	0.375	0.237	0.336	0.348
MolRA	**0.486**	**0.645**	**0.498**	**0.578**	0.590

MODEL	B-4↑	R-1↑	R-2↑	R-L↑	M↑
Molecular Captioning Task					
LLaMA2*	0.253	0.376	0.275	0.342	0.338
Mol-Instruct	0.171	0.331	0.203	0.289	0.271
HIGHT	0.397	0.582	0.414	0.518	0.525
InstructMol	0.371	0.566	0.394	0.502	0.509
LLaMo*	**0.485**	0.568	**0.476**	0.560	**0.602**
MolRA	0.460	**0.623**	0.469	**0.563**	0.565

MODEL	B-2↑	B-4↑	R-1↑	R-2↑	R-L↑
Experimental Procedure					
TextChemT5	0.541	0.406	0.615	0.403	0.564
MolT5-Large	0.545	0.410	0.625	0.409	0.572
Galactica	0.535	0.395	0.609	0.386	0.552
MolCA, Galac	0.549	0.415	0.625	0.404	0.570
ReactXT, Galac	**0.574**	**0.440**	**0.644**	**0.427**	**0.589**
MolRA	0.557	0.411	0.574	0.403	0.566

both general-purpose LLMs and existing molecule-oriented baselines.

Mol2Mol Tasks: Chemical Transformation Prediction. On tasks requiring structural reasoning, including forward reaction prediction, retrosynthesis, and reaction condition generation, our framework achieves consistently strong performance across Exact Match, edit distance, and fingerprint-based similarity metrics (Table 1). Compared to input-alignment-based models such as PRESTO (Park et al., 2024) and InstructMol (Cao et al., 2025), MolRA attains comparable or improved Exact Match scores while maintaining high chemical validity. These results indicate that direct parameter modulation enables effective structure-aware molecular reasoning for complex chemical

Table 3: Performance comparison across three different Mol2Num tasks. For yield regression, we report the $R^2 score$.

MODEL	Weight↓	LogP↓	TPSA↓
More `Mol2Num`			
LLaMA2	22.10 (96)	1.45 (95)	15.87 (92)
Vicuna v1.5	17.10 (93)	1.20 (92)	12.95 (93)
3D-MoLM(S)	14.79 (95)	0.66 (97)	9.71 (93)
3D-MoLM(G)	16.58 (92)	0.78 (95)	10.90 (90)
MolRA	**11.72 (100)**	**0.57 (100)**	**5.75 (100)**

MODEL	B-H↑	S-M ↑
Yield Prediction		
LLaMA2	-0.476	0.121
Vicuna v1.5	-0.131	0.151
PRESTO	0.944	0.652
MolRA	**0.956**	**0.684**

MODEL	**HOMO↓**	**LUMO↓**	**GAP↓**	**Avg.↓**
QM9 Property Prediction				
LLaMA2	0.737	0.864	0.515	0.7053
Mol-Instruct	0.0210	0.0210	0.0203	0.0207
HIGHT	0.0056	0.0065	0.0077	0.0066
InstructMol	0.0048	0.0050	0.0061	0.0050
UniMoT	0.0042	0.0047	0.0055	0.0048
MolRA	**0.0032**	**0.0037**	**0.0045**	**0.0038**

transformations, avoiding the structural degradation introduced by linear tokenization in prior approaches.

Mol2Text Tasks: Molecular Description Generation. For molecular description generation and captioning, our framework achieves competitive performance across BLEU, ROUGE, and METEOR metrics (Table 2). These results suggest that the molecular structural features encoded by the weight generator can be effectively integrated into the LLM to support language generation. By avoiding graph token expansion, MolRA preserves structural information during generation, enabling natural language descriptions that faithfully reflect molecular topology.

Mol2Num Tasks: Molecular Property Prediction. We evaluate MolRA on multiple Mol2Num benchmarks, including molecular property prediction on QM9, extended physicochemical property prediction, and yield regression (Table 3). Across molecular property prediction tasks, MolRA consistently achieves lower Mean Absolute Error (MAE) on most target properties, indicating improved accuracy in quantitative structure–property modeling. For reaction yield prediction, MolRA attains higher R^2 scores compared to input-alignment-based baselines, demonstrating its effectiveness in capturing structure-dependent numerical relationships.

5 Analysis

5.1 Analysis of the Computational Cost

To evaluate inference efficiency, we measured total GFLOPs and latency on the molecule captioning task using a diverse set of 200 molecules with sizes ranging from 0 to 300 atoms. As illustrated in Figure 3, input-alignment-based baselines (InstructMol, LLaMo, and PRESTO) exhibit a strong positive correlation between atom count and computational cost, with Pearson correlation coefficients (ρ) approaching 1.00, indicating that inference cost increases with molecular size. In contrast, MolRA maintains a nearly constant computational footprint

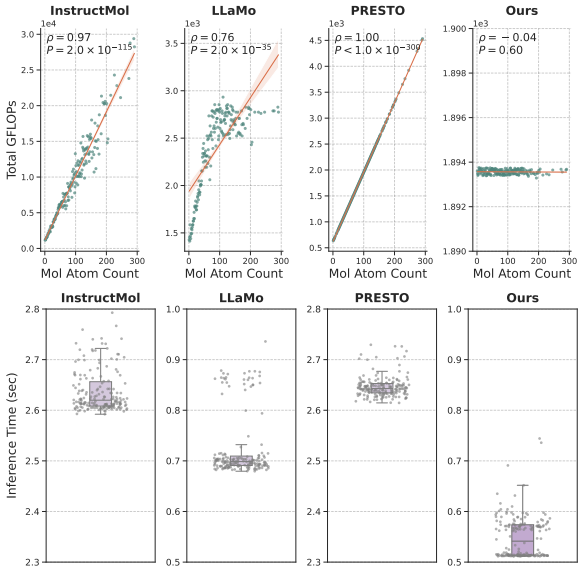


Figure 3: Computational cost analysis on the molecular description task. We compare Total GFLOPs (top row) and Inference Time (bottom row) across different molecular sizes (0-300 atoms).

across molecules of different sizes, showing a minimal correlation between atom count and GFLOPs ($\rho \approx -0.04$). These empirical observations suggest that MolRA avoids the computational escalation inherent to token-based molecular representations. A theoretical analysis of the computational complexity is provided in Appendix C.

5.2 Extensibility across LLM Backbones

As shown in Table 4, we evaluate MolRA on LLaMA 2-7B, Vicuna v1.5-7B, and Qwen 2.5-7B, where it consistently outperforms baselines like LoRA and full fine-tuning (e.g., surpassing fine-tuning by substantial margins on LLaMA 2). Notably, Vicuna v1.5 exhibits the superior performance. We attribute this to its robust instruction-following alignment, which synergizes effectively with our dynamic weight injection to better articulate molecular reasoning cues despite the frozen backbone. These results confirm that MolRA is a transferable strategy that benefits from strong base model alignment.

Table 4: Extensibility analysis of MolRA across different LLM backbones compared to LoRA and full fine-tuning on molecular forward reaction prediction tasks.

Backbone	Training Setting	BLEU-2↑	Levenshtein↓	FTS↑	Validity↑
LLaMA 2-7B	Infer-only (Base)	0.036	46.844	0.017	0.041
	+ LoRA	0.764	15.532	0.703	1.000
	+ Full FT	0.821	13.248	0.728	1.000
	+ MolRA (Ours)	**0.941**	**10.983**	**0.787**	**1.000**
Vicuna v1.5-7B	Infer-only (Base)	0.034	45.654	0.021	0.038
	+ LoRA	0.775	15.502	0.698	1.000
	+ Full FT	0.828	13.321	0.728	1.000
	+ MolRA (Ours)	**0.960**	**10.720**	**0.780**	**1.000**
Qwen 2.5-7B	Infer-only (Base)	0.015	50.524	0.015	0.029
	+ LoRA	0.525	19.531	0.648	0.992
	+ Full FT	0.748	15.701	0.685	0.998
	+ MolRA (Ours)	**0.928**	**12.373**	**0.742**	**0.998**

5.3 Analysis of Modulation Placement

We conduct an ablation study to examine effective injection locations within a standard Vicuna decoder block. Candidate targets include the multi-head attention projections—query (q), key (k), value (v), output projection (o)—as well as the feed-forward network (f). As shown in Figure 4, injecting molecule-aware weights into all components (`qkvof`) yields the best overall performance. This result indicates that jointly modulating both the self-attention and feed-forward sublayers of Vicuna v1.5-7B is beneficial for effective structural conditioning.

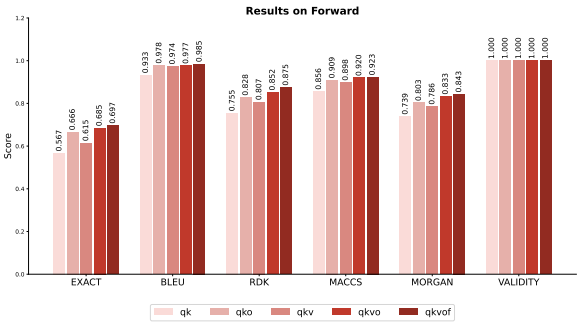
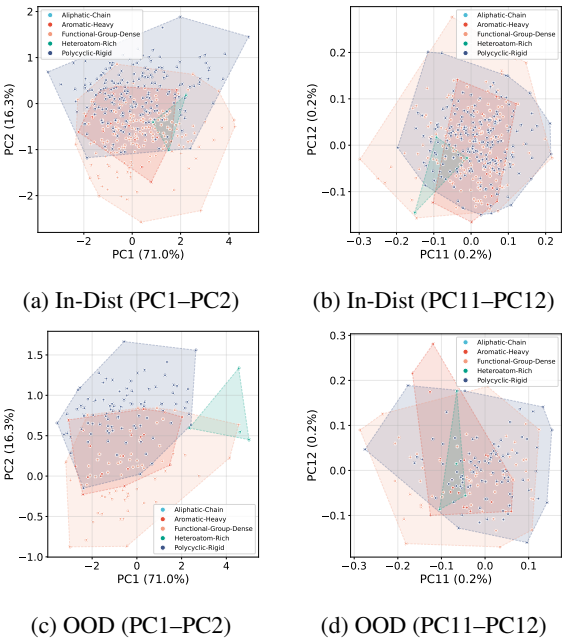


Figure 4: Ablation study on the injection targets for the molecule-aware weights. Targets include the self-attention projections—query (q), key (k), value (v), output (o), and the feed-forward network (f).

5.4 Analysis of Parameter Modulations

We visualize PCA projections of the generated ΔW across five chemical classes. Unlike static LoRA, MolRA produces instance-specific parameter updates that reflect molecular structural differences, as illustrated in Figure 5. The first two principal components (PC1–PC2), accounting for 87.3% of the variance, exhibit clear clustering patterns associated with broad structural characteristics (e.g., polycyclic rigid versus aliphatic flexible molecules). Similar geometric distributions



(a) In-Dist (PC1–PC2) (b) In-Dist (PC11–PC12)

(c) OOD (PC1–PC2) (d) OOD (PC11–PC12)

Figure 5: PCA Visualization of Learned Parameter Modulations (ΔW). The geometric consistency between In-Dist and OOD settings across primary (PC1–PC2) and lower-order (PC11–PC12) components demonstrates robust structural understanding.

are observed under both in-distribution and out-of-distribution settings, suggesting that the learned parameter modulations generalize beyond individual samples. In addition, lower-variance components capture more fine-grained variations, indicating a hierarchical organization in which dominant directions encode coarse structural information.

6 Conclusion

In this paper, we introduced MolRA, a parameter space modulation framework for molecular multimodal large language models. By shifting molecular structural modeling from input space alignment to parameter space modulation, MolRA avoids sequence inflation and structural fragmentation in graph tokenization. Through the molecule adaptive weight generator, molecular graphs are transformed into low-rank parameter modulations injected into a frozen LLM, enabling structural reasoning without expanding the input sequence. Extensive experiments show that MolRA achieves strong predictive performance and computational efficiency. Overall, this work highlights parameter-space modulation as a promising direction for scalable multimodal integration in scientific LLMs.

Limitations

Despite the promising performance of MolRA, two primary limitations remain. First, our framework currently relies solely on 2D molecular topologies, potentially overlooking stereochemical nuances crucial for spatial tasks. Future iterations will explore integrating 3D conformers or multimodal data to enrich molecular representation. Second, we utilize general-domain LLMs (e.g., Vicuna, LLaMA, Qwen) as the frozen backbone. While this ensures instruction-following proficiency, the inherent lack of specialized scientific knowledge in the base model may restrict the upper bound of chemical reasoning. Extending MolRA to larger, domain-specific foundation models constitutes a key direction for future work.

Ethics Statement

Our work complies with the ACL Ethics Policy. The proposed MolRA is a general parameter-space modulation method for molecular multimodal large language models and does not involve human subjects or sensitive data. All datasets and models used are publicly available. We do not foresee any significant ethical concerns associated with this work, nor additional chemical safety risks.

References

He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. 2025. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. In *COLING*.

He Cao, Yanjun Shao, Zhiyuan Liu, Zijing Liu, Xiangru Tang, Yuan Yao, and Yu Li. 2024. Presto: Progressive pretraining enhances synthetic chemistry outcomes. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10197–10224.

Yongqiang Chen, Quanming Yao, Juzheng Zhang, James Cheng, and Yatao Bian. 2024. Hight: Hierarchical graph tokenization for graph-language alignment. *arXiv preprint arXiv:2406.14021*.

Zhe Chen, Zhe Fang, Wenhao Tian, Zhaoguang Long, Changzhi Sun, Yuefeng Chen, Hao Yuan, Honglin Li, and Man Lan. 2025. Reactgpt: Understanding of chemical reactions via in-context tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 84–92.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. 2023. Unifying molecular and textual representations via multi-task language modelling. In *International Conference on Machine Learning*, pages 6140–6157. PMLR.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. 2022. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Shuhan Guo, Yatao Bian, Ruibing Wang, Nan Yin, Zhen Wang, and Quanming Yao. 2025. Unified molecule-text language model with discrete token representation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 9205–9213.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, and 1 others. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Chengxin Hu, Hao Li, Yihe Yuan, Zezheng Song, Chenyang Zhao, and Haixin Wang. Omni-mol: Multitask molecular model for any-to-any modalities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. 2022a. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. 2022b. Chemformer: a

pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022.

Dongki Kim, Wonbin Lee, and Sung Ju Hwang. 2025. Mol-llama: Towards general understanding of molecules in large molecular language model. *CoRR*.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, and 1 others. 2023. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380.

Khiem Le, Zhichun Guo, Kaiwen Dong, Xiaobao Huang, Bozhao Nan, Roshni Iyer, Xiangliang Zhang, Olaf Wiest, Wei Wang, and Nitesh V Chawla. 2024. Molx: Enhancing large language models for molecular learning with a multi-modal extension. *arXiv preprint arXiv:2406.06777*.

Chanhui Lee, Yuheon Song, YongJun Jeong, Hanbum Ko, Rodrigo Hormazabal, Sehui Han, Kyunghoon Bae, Sungbin Lim, and Sungwoong Kim. 2025. Mollm: Generalist molecular llm with improved graph utilization. *arXiv e-prints*, pages arXiv–2502.

Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. Towards 3d molecule-text interpretation in language models. In *The Twelfth International Conference on Learning Representations*.

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

Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. 2023b. Multimodal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. a. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter.

Zhiyuan Liu, Yaorui Shi, An Zhang, Sihang Li, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. b. Reactxt: Understanding molecular "reaction-ship" via reaction-contextualized molecule-text pretraining.

Micha Livne, Zulfat Miftahutdinov, Elena Tutubalina, Maksim Kuznetsov, Daniil Polykovskiy, Annika Brundyn, Aastha Jhunjhunwala, Anthony Costa, Alex Aliper, Alán Aspuru-Guzik, and 1 others. 2024. nach0: multimodal natural and chemical languages foundation model. *Chemical Science*, 15(22):8380–8389.

Daniel Lowe. 2017. Chemical reactions from us patents (1976-sep2016). *(No Title)*.

Jieyu Lu and Yingkai Zhang. 2022. Unified deep learning model for multitask reaction predictions with explanation. *Journal of chemical information and modeling*, 62(6):1376–1387.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023. Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. *arXiv preprint arXiv:2308.09442*.

Jinyoung Park, Minseong Bae, Dohwan Ko, and Hyunwoo J Kim. 2024. Llamo: Large language model-based molecular graph assistant. *Advances in Neural Information Processing Systems*, 37:131972–132000.

Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. 2024a. Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1216–1240.

Qizhi Pei, Lijun Wu, Kaiyuan Gao, Jinhua Zhu, and Rui Yan. 2024b. 3d-molt5: Towards unified 3d molecule-text modeling with 3d molecular tokenization. *CoRR*.

Yujie Qian, Zhening Li, Zhengkai Tu, Connor Coley, and Regina Barzilay. 2023. Predictive chemistry augmented with text retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12731–12745.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. 2020. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571.

Philippe Schwaller, Alain C Vaucher, Teodoro Laino, and Jean-Louis Reymond. 2021. Prediction of chemical reaction yields using deep learning. *Machine learning: science and technology*, 2(1):015016.

Bonggun Shin, Sungsoo Park, Keunsoo Kang, and Joyce C Ho. 2019. Self-attention based molecule representation for predicting drug-target interaction. In *Machine learning for healthcare conference*, pages 230–248. PMLR.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15638–15650.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Duong Thanh Tran, Nguyen Doan Hieu Nguyen, Nhat Truong Pham, Rajan Rakkiyappan, Rajendra Karki, and Balachandran Manavalan. 2025. Xmolcap: Advancing molecular captioning through multimodal fusion and explainable graph neural networks. *IEEE Journal of Biomedical and Health Informatics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yue Wan, Chang-Yu Hsieh, Ben Liao, and Shengyu Zhang. 2022. Retroformer: Pushing the limits of end-to-end retrosynthesis transformer. In *International Conference on Machine Learning*, pages 22475–22490. PMLR.

Botao Yu, Frazier N Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv:2402.09391*.

Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2022. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1):862.

Juexiao Zhou, Xiaonan He, Liyuan Sun, Jiannan Xu, Xiuying Chen, Yuetan Chu, Longxi Zhou, Xingyu Liao, Bin Zhang, Shawn Afvari, and 1 others. 2024. Pre-trained multimodal large language model enhances dermatological diagnosis using skingpt-4. *Nature Communications*, 15(1):5649.

Appendix

A Detailed Experimental Settings

A.1 Details of Benchmarks for Evaluation

Molecular Description Generation. This task evaluates a model's ability to generate informative text descriptions for a given molecule. The descriptions should cover key aspects such as chemical properties, functional groups, and biological roles. We use two main datasets for this task. (1) ChEBI-20 provides rich, detailed descriptions for molecules, focusing on biochemical entities. (2) The PubChem Description Q&A dataset is structured in a question-and-answer format, testing the model's ability to retrieve specific factual information about a molecule.

Forward Reaction Prediction. In this task, the model predicts the likely product of a chemical reaction given the reactants and reagents. The model receives input as SELFIES strings representing the starting materials and must generate the SELFIES string for the resulting product. Our evaluation is based on the widely used USPTO dataset, which contains a large collection of chemical reactions from U.S. patents.

Retrosynthesis. Retrosynthesis is the reverse of forward reaction prediction. Given a target product molecule, the model's goal is to identify and generate the SELFIES strings of the necessary reactants for its synthesis. This task challenges the model's understanding of chemical reaction pathways. We use the USPTO 500K dataset, a large-scale benchmark specifically curated for single-step retrosynthesis.

Reagent Prediction. This task assesses the model's ability to identify the correct reagents needed to facilitate a specific chemical transformation. The model is provided with both the reactants and the final product, and it must predict the appropriate reagent(s). For this task, we utilize the USPTO 500 MT dataset.

Property Prediction (Regression). This task measures the model's performance in predicting quantitative molecular properties. We focus on quantum mechanical properties, specifically predicting the energy of the Highest Occupied Molecular Orbital (HOMO) and the Lowest Unoccupied Molecular Orbital (LUMO). The model takes a molecule's SELFIES representation as input and outputs a continuous numerical value for the target property. We use the QM9 dataset for this evaluation.

A.2 Details of Evaluation Metrics

We use a range of established metrics to evaluate our model's performance across different tasks. The direction of the arrow (↑/↓) indicates whether a higher or lower score is better.

Chemical Reaction Prediction. For tasks involving SELFIES generation (Forward Reaction Prediction, Retrosynthesis, Reagent Prediction, Catalyst Prediction, Solvent Prediction), we use the following metrics:

- **Exact Match (↑):** The percentage of predictions where the canonicalized SELFIES string of the generated molecule is identical to the ground truth. This is a strict measure of accuracy.
- **BLEU (↑):** Measures the n-gram overlap between the generated and reference SELFIES strings. Although typically used for natural language, it serves as a proxy for textual similarity in SELFIES generation.
- **Levenshtein Score (↓):** Measures the minimum number of single-character edits (insertions, deletions, substitutions) needed to transform the predicted SELFIES into the ground-truth SELFIES. It quantifies the textual difference between the strings.
- **Structural Similarity (↑):** We assess the structural likeness between predicted and true molecules using Tanimoto similarity calculated from different molecular fingerprints: **MACCS**, **RDKit**, and **Morgan** fingerprints. A higher score indicates greater structural resemblance.
- **Validity (↑):** The percentage of generated SELFIES strings that represent chemically valid and parsable molecules. This is a fundamental check for the model's chemical correctness.

Molecule Description Generation. To assess the quality of the generated text, we employ a comprehensive set of standard metrics that compare the predicted descriptions against reference texts:

Table 5: Statistics of datasets utilized in the MolRA training pipeline. Stage 1 focuses on molecule-text alignment, while Stage 2 and downstream tasks cover diverse chemical reasoning paradigms.

Task Type	Data Source	# Train	# Valid	# Test	# Total
Pretraining Stage 1: Molecule-Text Alignment					
Molecule Captioning	PubChem (Kim et al., 2023)	326,675	–	–	326,675
Downstream Paradigm I: Mol2Mol (Structural Reasoning)					
Forward Prediction	USPTO (Lu and Zhang, 2022; Yu et al., 2024)	124,384	–	1,000	125,384
Retrosynthesis	USPTO (Fang et al.)	124,384	–	1,000	125,384
Reagent Prediction	TextReact (Qian et al., 2023)	57,162	6,216	6,378	69,756
Catalyst Prediction	ChemLLM (Guo et al., 2023)	10,232	1,059	1,015	12,306
Solvent Prediction	Mol-Instruction (Fang et al.)	70,988	7,694	7,793	86,475
Downstream Paradigm II: Mol2Text (Cross-modal Translation)					
Description Q&A	PubChem (Kim et al., 2023)	56,885	1,000	1,000	328,675
Experimental Procedure	OpenReaction(Lowe, 2017)	120,811	–	26,977	147,788
Downstream Paradigm III: Mol2Num (Quantitative Reasoning)					
QM9 Property Prediction	QM9 (Fang et al.)	110,000	10,000	10,831	130,831
Yield Prediction (Buchwald)	YieldBERT (Schwaller et al., 2021)	3,855	–	100	3,955
Yield Prediction (Suzuki)	YieldBERT (Schwaller et al., 2021)	5,660	–	100	5,760

- **BLEU-4 (B-4) (↑):** Measures the precision of n-grams (up to 4-grams), evaluating the fluency and adequacy of the generated text by checking for co-occurring word sequences.
- **ROUGE (R) (↑):** We report three variants to measure recall. **ROUGE-1 (R-1)** and **ROUGE-2 (R-2)** compute the overlap of unigrams and bigrams, respectively. **ROUGE-L (R-L)** is based on the Longest Common Subsequence (LCS) and captures sentence-level structural similarity.
- **METEOR (M) (↑):** A more advanced metric that creates an alignment between the generated and reference texts, considering not only exact matches but also stems and synonyms for a more semantically-aware evaluation.

Property Prediction (Regression). For the quantitative prediction of quantum properties, we evaluate the model's accuracy on HOMO energy, LUMO energy, and the HOMO-LUMO Gap.

- **Mean Absolute Error (MAE) (↓):** We report the MAE for each property (**HOMO**, **LUMO**, **GAP**) to measure the specific prediction error for each task. We also compute the **Average MAE (Avg. MAE)**, which is the mean of these three MAE values, to provide a single, overall performance score.
- **R-squared (R^2) (↑):** As a complementary metric, the coefficient of determination measures the proportion of variance in the target properties that is explained by the model, serving as a statistical indicator of the goodness of fit between the predicted values and the actual ground truth.

A.3 Implementation Details

MolRA is built upon the frozen Vicuna-7B backbone, utilizing a frozen MoleculeSTM as the GNN-based graph encoder with a 300-dimensional embedding space. The primary trainable component is our proposed Adaptive Weight Generator (AWGen), which is specifically designed to produce instance-specific updates through a query-based distillation mechanism. The architecture of AWGen consists of $N = 8$ decoder blocks and $k = 4$ learnable molecular queries. The generated low rank updates, with a rank of $r = 64$ and $\alpha = 64$, are injected into the self-attention matrices ($\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$) as well as the feed forward network (FFN) projections ($\mathbf{f}$). To ensure training stability, the linear projection heads within AWGen are zero-initialized. We employ the AdamW optimizer to minimize the standard cross-entropy loss for text generation. All models are trained in bfloat16 precision on 8 NVIDIA A800 GPUs. The detailed training configurations for each molecular task category are summarized in Table 6.

B Additional Ablations

B.1 Analysis of Modulation Behavior

We analyze the impact of the depth of the Adaptive Weight Generator by varying the number of transformer blocks (N). As shown in Table 7, performance consistently improves as N increases from 2 to 8 across all metrics. This suggests that shallow configurations lack sufficient expressive ca-

Table 6: Implementation details for our MolRA model.

Configuration	MolRA
Molecule Encoder	MoleculeSTM
Molecule Emb. Dim.	300
LLM Base	Vicuna-7B
MolRA Injection Targets	q, k, v, o, f
MolRA-Rank (r)	64
MolRA-Alpha (α)	64
MAW-Gen Blocks (N)	8
Molecular Queries (k)	4
Training Hyperparameters	
Epochs	15
Optimizer	AdamW
Batch Size	96 (Global)
Learning Rate	2e-5
LR Scheduler	Cosine
Warm-up	3% of total steps
Weight Decay	0.0
Precision	bfloat16
GPU Usage	8 NVIDIA A800s

Table 7: Effect of the number of transformer blocks (N) in the Adaptive Weight Generator . Performance is reported on forward reaction prediction and retrosynthesis tasks.

	Forward		Retrosynthesis	
Blocks (N)	**Exact↑**	**BLEU↑**	**Exact↑**	**BLEU↑**
2	0.663	0.967	0.493	0.928
4	0.645	0.969	0.508	0.944
8	**0.697**	**0.985**	**0.533**	**0.957**
16	0.689	0.980	0.530	0.955

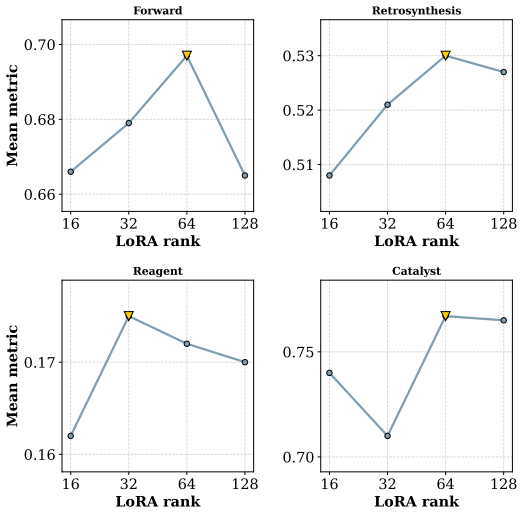


Figure 6: Impact of MolRA molecule-aware weights' rank on task performance. Mean performance metrics averaged over four chemically diverse tasks with rank $r \in \{16, 32, 64, 128\}$. Yellow triangles denote the best-performing rank for each metric.

pacity to fully capture molecular structural cues. When increasing N beyond 8, the performance gain saturates or slightly degrades, likely due to over-parameterization. Therefore, we adopt $N = 8$ as the default setting, which achieves the best overall performance on both forward prediction and retrosynthesis tasks.

B.2 Hyperparameter Sensitivity of MolRA's Rank

We investigate the impact of the rank r of the generated molecule-aware weights on the model performance across four diverse chemical reasoning tasks: Forward Reaction, Retrosynthesis, Reagent Prediction, and Catalyst Prediction. As illustrated in Figure 6, we observe a consistent trend where performance peaks at $r = 64$ across all metrics. Specifically, lower ranks ($r = 16, 32$) appear to lack sufficient capacity to encode complex molecular features, leading to underfitting; conversely, increasing the rank to 128 yields diminishing returns or slight degradation, likely stemming from overfitting or optimization difficulties. Based on these empirical findings, we adopt $r = 64$ as the default configuration for MolRA, striking an optimal balance between representational expressiveness and parameter efficiency.

C Theoretical Efficiency Analysis of MolRA

By maintaining zero graph tokens in the LLM input, MolRA decouples the computational cost of the backbone model from molecular complexity. We analyze the resulting efficiency gains through a theoretical FLOPs comparison, which is consistent with the empirical inference efficiency results reported in the main paper. For conventional molecular MLLMs based on input-space alignment (e.g., PRESTO), the computational complexity of a transformer layer can be expressed as $24(L+C)dh^2+4(L+C)^2dh$, where L denotes the number of graph tokens, C the number of text tokens, d the hidden dimension, and h the number of attention heads. The quadratic term arises from self-

attention over the combined graph–text sequence. In contrast, MolRA eliminates graph tokens from the input sequence, reducing the backbone complexity to $24Cdh^2 + 4C^2dh$. The additional cost introduced by MolRA corresponds to a one-time generation of low-rank molecule-aware weights, with complexity $24krh^2$, which is independent of molecular size and negligible compared to the backbone computation.

D Prompts and Case Studies

This section details the specific prompt templates and input-output formats used across various downstream tasks. For all tasks, the molecular graph information is encoded via the MolRA stream to modulate the LLM parameters (ΔW), while the textual instructions follow the Vicuna chat template.

MolRA Chat Template Architecture The MolRA architecture employs a dual-stream mechanism to strictly separate parameter modulation from textual interaction. Unlike standard multimodal LLMs that inject visual tokens into the context window, MolRA utilizes Stream A for parameter modulation, where the molecular graph is processed by the encoder and AW-Gen module to generate dynamic weights (ΔW). Stream B handles the textual interaction using the standard Vicuna chat template. As illustrated below, the `<molecule>` token acts solely as a sentinel marker for alignment during training and is not tokenized into the context window, preventing context pollution.

Algorithm 1: Pseudocode of MolRA

Input: Training dataset $\mathcal{D} = \{(\mathcal{G}_i, I_i, A_i)\}_{i=1}^{N}$, Frozen GNN encoder $g(\cdot)$, Frozen LLM with parameters θ_{LLM}
Output: Trained AW-Gen parameters ψ

Initialize:
Molecular queries $\mathbf{Q}_{\text{learn}} \in \mathbb{R}^{k \times d_{\text{model}}}$
Transformer decoder with N layers
Projection layer $\mathbf{W}_{\text{FC}} \in \mathbb{R}^{d_{\text{model}} \times (d_{\text{llm}} \cdot r)}$
Shared projection $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{r \times d_{\text{llm}}}$
Components to modulate $\mathcal{C} = \{W_q, W_k, W_v, W_o, W_f\}$ across L_{adapt} layers

for *epoch* $= 1$ **to** E **do**
  for *batch* $(\mathcal{G}, I, A)$ ***in*** $\mathcal{D}$ **do**
    `// Step 1: Encode molecular graph`
    $\mathbf{H}_{\mathcal{G}} \leftarrow g(\mathcal{G})$
    `// Step 2: Cross-attention distillation`
    $\mathbf{Q} \leftarrow \mathbf{Q}_{\text{learn}}$
    for $l = 1$ **to** N **do**
      $\mathbf{Q} \leftarrow \text{SelfAttention}(\mathbf{Q})$
      $\mathbf{Q} \leftarrow \text{CrossAttention}(\mathbf{Q}, \mathbf{H}_{\mathcal{G}})$
      $\mathbf{Q} \leftarrow \text{FFN}(\mathbf{Q})$
    $\mathbf{Q}_{\text{out}} \leftarrow \mathbf{Q}$; `// Distilled queries`
    `// Step 3: Generate low-rank weight updates`
    for *layer* $i = 1$ **to** L_{adapt} **do**
      for *component* $c \in \mathcal{C}$ **do**
        $\mathbf{q}_{i,c} \leftarrow \mathbf{Q}_{\text{out}}[i \cdot |\mathcal{C}| + c]$; `// Select query`
        $\Delta\mathbf{A}_{i,c} \leftarrow \text{Reshape}(\mathbf{q}_{i,c} \cdot \mathbf{W}_{\text{FC}})$; `// Shape:` $d_{\text{llm}} \times r$
        $\mathbf{W}_{\text{mol},i,c} \leftarrow \Delta\mathbf{A}_{i,c} \cdot \mathbf{W}_{\text{proj}}$; `// Shape:` $d_{\text{llm}} \times d_{\text{llm}}$
    `// Step 4: Inject weights into frozen LLM`
    for *layer* $i = 1$ **to** L_{adapt} **do**
      for *component* $c \in \mathcal{C}$ **do**
        $\hat{\mathbf{W}}_{i,c} \leftarrow \mathbf{W}_{i,c} + \mathbf{W}_{\text{mol},i,c}$; `// Parameter modulation`
    `// Step 5: Forward pass with adapted LLM`
    $\hat{A} \leftarrow \text{LLM}_{\text{adapted}}(I; \{\hat{\mathbf{W}}_{i,c}\})$; `// Only text input`
    `// Step 6: Compute loss and update AW-Gen`
    $\mathcal{L} \leftarrow -\sum_{t=1}^{|A|} \log P(a_t | a_{<t}, I, \mathcal{G}; \psi)$
    Update $\psi = \{\mathbf{Q}_{\text{learn}}, \mathbf{W}_{\text{FC}}, \mathbf{W}_{\text{proj}}\}$ using $\nabla\mathcal{L}$
return *Trained AW-Gen with parameters* ψ

MolRA's Chat Template

Stream A: Parameter Modulation (via MolRA)

$\mathcal{G}_{molecule} \xrightarrow{\text{Encoder + AW-Gen}} \Delta W$ **(Dynamic Weights)** $\xrightarrow{\text{Inject}} LLM_\theta$

Stream B: Textual Interaction

SYSTEM: A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions.

USER: `<molecule><\n><instruction> <sep>`

ASSISTANT: `<prediction> </s>`

**Note: `<molecule>` acts as a sentinel marker for alignment but is not tokenized into the context window. Instead, the molecule graph modulates LLM_θ parameters via ΔW.*

Figure 7: The dual-stream architecture of MolRA. Unlike standard multimodal LLMs, MolRA separates parameter modulation (Stream A) from textual interaction (Stream B), ensuring strict adherence to the Vicuna chat template without context window pollution.

Case Study: Forward Reaction Prediction

[MolRA Stream] Reactants → Dynamic Weights:

`[C][O][C][=Branch1][C][=O][C][S][N][=N][C][=Ring1][Branch1][C][C][C][Ring1][Ring1].[C][O].[Na+1].[OH1-1]` $\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:

Predict the main product of the chemical reaction corresponding to the provided molecular graph structure.

MolRA Output (Generated with modulated weights):

`[O][=C][Branch1][C][O][C][S][N][=N][C][=Ring1][Branch1][C][C][C][Ring1][Ring1]`

Figure 8: Case study on the **Forward Reaction Prediction** task. MolRA successfully encodes the reactants into dynamic weights $\mathbf{W}_{\text{mol}}$, guiding the LLM to generate the correct product SELFIES sequence.

Case Study: Retrosynthesis Prediction

[MolRA Stream] Target Product → Dynamic Weights:

`[O][=C][Branch1][=Branch1][C][C][C][C][Br][N][C][C][C][C][C][O]`

$\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:

Given the following product, please provide possible reactants.

MolRA Output (Predicted Reactants):

`[N][C][C][C][C][C][O].[O][=C][Branch1][C][Cl][C][C][C][C][Br]`

Figure 9: Case study on Retrosynthesis Prediction. Given the target product structure, MolRA encodes the molecular graph into dynamic weights $\mathbf{W}_{\text{mol}}$. The model successfully performs the retrosynthetic analysis, identifying the correct amine and acyl chloride precursors required to synthesize the target molecule.

Case Study: Reagent Prediction

[MolRA Stream] Reaction (Reactants → Product) → Weights:

`[C][C][C][=Branch1][C][=O][Cl].[O][C][=C][C][=C][C][Branch1][C][F][=C][Ring1][#Branch1]»[C][C][C][=Branch1][C][=O][O][C][=C][C][=C][C][Branch1][C][F][=C][Ring1][#Branch1]`

$\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:

From the provided chemical reaction, propose some possible reagents that could have been used.

MolRA Output (Proposed Reagents):

`[Cl][C][Cl].[C][=C][C][=N][C][=C][Ring1][=Branch1]`

Figure 10: Case study on **Reagent Prediction**. Given the transformation from reactants to product, MolRA infers the missing chemical context, correctly identifying Pyridine (base) and DCM (solvent) required to facilitate the acylation reaction.

Case Study: Catalyst Prediction

[MolRA Stream] Reaction (Reactant → Product) → Weights:

[C][C][Branch1][C][C][N][C][C][C][C][N][Branch1][#Branch2][C][C][=C][C][=C][C][=C][Ring1][=Branch1][C][Ring1][N]»
[C][C][Branch1][C][C][N][C][C][C][C][N][C][Ring1][Branch1]

$\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
Can you provide potential catalyst for the following chemical reaction?

MolRA Output (Predicted Catalyst):

[Pd]

Figure 11: Case study on **Catalyst Prediction**. MolRA analyzes the structural changes between the reactant and product encoded in $\mathbf{W}_{\text{mol}}$—specifically the cleavage of the benzyl-nitrogen bond—and predicts Palladium as the correct metal catalyst.

Case Study: Solvent Prediction

[MolRA Stream] Reaction Context → Dynamic Weights:

[C][C][=Branch1][C][=O][O][C][Branch1][C][C][=O].[O][=C][N][C][=C][C][=C][Branch1][C][O][C][=C][C][=C][C][=C][C][=C][Ring1][O][Ring1][=Branch1]»
[C][C][=Branch1][C][=O][O][C][=C][C][=C][C][=C][C][=C][C][Ring1][=Branch1][=C][Ring1][#Branch2][C][=C][N][C][=O]

$\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
Given the following reaction, what are some possible solvents that could have been utilized?

MolRA Output (Predicted Solvent):

[C][=C][C][=N][C][=C][Ring1][=Branch1]

Figure 12: Case study on **Solvent Prediction**. MolRA analyzes the reaction transformation (from anhydride/amine to amide/ester product) encoded in $\mathbf{W}_{\text{mol}}$ and correctly identifies Pyridine as the optimal solvent system.

Case Study: Molecular Property Prediction (Molecular Weight)

[MolRA Stream] Target Molecule → Dynamic Weights:

`[C][N][Branch1][#Branch1][C][C][=Branch1][C][=O][N][C][=Branch1][C][=O][N][Branch1][Ring2][C][C][Cl][N][=O]` $\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
Please provide me with the Molecular Weight value of this molecule. Determine the Molecular Weight value of this molecule. If uncertain, provide an estimate. Respond with the numerical value only.

MolRA Output:
The Molecular Weight for the input molecule is 221.03 g/mol.

Figure 13: Case study on **Molecular Weight Prediction**. This task validates the model's ability to recognize atomic types and quantities. MolRA injects this structural information via $\mathbf{W}_{\text{mol}}$, allowing the LLM to output the precise mass (222.63 g/mol) in a natural language format.

Case Study: Molecular Property Prediction (TPSA)

[MolRA Stream] Target Molecule → Dynamic Weights:

`[C][=C][C][=C][C][=Branch1][Ring2][=C][Ring1][=Branch1][C][=C][C][=Branch1][Ring2] [=N][Ring1][#Branch1][C][=C][C][=C][Ring1][#Branch1][N]` $\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
Please provide me with the Topological Polar Surface Area value of this molecule. Determine the Topological Polar Surface Area value of this molecule. If uncertain, provide an estimate. Respond with the numerical value only.

MolRA Output:
The Topological Polar Surface Area for the input molecule is 38.90 Å^2.

Figure 14: Case study on **TPSA Prediction**. This task requires the model to quantify the polar surface area, a critical descriptor for drug transport properties. MolRA accurately predicts the value (38.90Å^2) and formats it within a natural language response.

Case Study: Molecular Property Prediction (LogP)

[MolRA Stream] Target Molecule → Dynamic Weights:

`[C][N][Branch1][#Branch1][C][C][=Branch1][C][=O][N][C][=Branch1][C][=O][N][Branch1][Ring2][C][C][Cl][N][=O]` $\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
I would like to know the LogP of this molecule, could you please provide it? If uncertain, provide an estimate. Respond with the numerical value only.

MolRA Output:
The LogP for the input molecule is -1.00.

Figure 15: Case study on **LogP Prediction**. MolRA evaluates the molecule's lipophilicity/hydrophilicity ratio by processing its structural graph into dynamic weights. The model successfully predicts the partition coefficient, a crucial parameter for assessing drug-likeness and bioavailability.

Case Study: Molecular Property Prediction (LUMO)

[MolRA Stream] Target Molecule → Dynamic Weights:

`[O][C][C][C][C][C][C][Ring1][Branch1][C][Ring1][=Branch1][O][Ring1][Branch1]` $\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
Please provide the LUMO energy value for this molecule.

MolRA Output (Predicted Value):
`0.0711`

(Analysis: Demonstrating high-precision regression capability, MolRA utilizes the molecule-aware weights to predict the Lowest Unoccupied Molecular Orbital (LUMO) energy directly as a numerical token.)

Figure 16: Case study on the **Property Prediction** task. MolRA is not limited to text generation; it can also perform regression tasks. Here, the structural information injected via $\mathbf{W}_{\text{mol}}$ enables the LLM to accurately predict quantum chemical properties like LUMO energy.

Case Study: Reaction Yield Prediction

[MolRA Stream] Full Reaction Sequence → Dynamic Weights:

`CC(=O)O~CC(=O)O~[Pd].[C][C][C][=C][C][=C][C][C][C][=C][C][=C][C]...`

» `[C][C][=C][C][=C][C][Branch1][=C][C][=N][N]...` $\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
Please provide the yield ratio of the reaction.

MolRA Output (Predicted Yield Ratio):
`0.363`

Figure 17: Case study on the **Yields Regression** task. The input includes the full reaction context (reactants, reagents, and product). MolRA modulates the LLM to predict a scalar value representing the reaction efficiency (yield ratio).

Case Study: Molecular Description QA

[MolRA Stream] Target Molecule → Dynamic Weights:

`[C][Branch1][=Branch1][C][=Branch1][C][=O][O][N][C][=Branch1][C][=O][N]`
$\xrightarrow{\text{AW-Gen}}$ $\mathbf{W}_{\text{mol}}$

Text Instruction:
What functional groups are present in N-carbamoylglycine?

MolRA Output (Description):
N-carbamoylglycine contains a carbamide functional group (-CONH2) and a carboxyl group (-COOH).

Figure 18: Case study on the **DescriptionQA** task. MolRA extracts structural features via the parameter stream ($\mathbf{W}_{\text{mol}}$) to answer open-ended questions about the molecule's chemical properties, such as identifying functional groups.

Case Study: Experimental Procedure Prediction

[MolRA Stream] Reaction Components → Dynamic Weights:

Reactants (1): `[C][C][C][S][=Branch1][C][=O][=Branch1][C][=O][N][C]...`

Solvents (2): `[C][C][C][O][C][Ring1][Branch1]`

Product (-1): `[C][C][C][S][=Branch1][C][=O][=Branch1][C][=O][N][C]...`

$\xrightarrow{\text{AW-Gen}} \mathbf{W}_{\text{mol}}$

Text Instruction:

Using the given chemical species (reactants, product, solvents, catalysts), outline the detailed experimental procedure.

MolRA Output (Action Sequence):

A solution was prepared by dissolving compound 1 (21.8 g, 76 mmol) in solvent 2 (100 mL). Reagent 3 (2.45 equiv.) was then introduced, and the mixture was stirred at ambient temperature. After completion, the reaction mixture was concentrated and adjusted to neutral pH using aqueous HCl. The resulting solids were collected by filtration, washed with water, and the filtrate was dried over a drying agent. Removal of the drying agent followed by filtration afforded product −1 (20.10 g).

Figure 19: Case study on **Experimental Procedure Prediction**. In this task, MolRA translates the molecular structures of reactants and reagents into a sequential, executable laboratory recipe, bridging the gap between chemical entities and operational actions.