oMeBench: Towards Robust Benchmarking of LLMs in Organic Mechanism Elucidation and Reasoning

Anonymous ACL submission

Abstract

Organic reaction mechanisms are the stepwise elementary reactions by which reactants form intermediates and products, and are fundamental to understanding chemical reactivity and designing new molecules and reactions. Although large language models (LLMs) have shown promise in understanding chemical tasks such as synthesis design, it is unclear to what extent this reflects genuine chemical reasoning capabilities, i.e., the ability to generate valid intermediates, maintain chemical consistency, and follow logically coherent multi-step pathways. We address this by introducing **oMeBench**, the first large-scale, expert-curated benchmark for organic mechanism reasoning in organic chemistry. It comprises over 10,000 annotated mechanistic steps with intermediates, type labels, and difficulty ratings. Furthermore, to evaluate LLM capability more precisely and enable fine-grained scoring, we propose **oMeS**, a dynamic evaluation framework that combines step-level logic and chemical similarity. We analyze the performance of state-of-the-art LLMs, and our results show that although current models display promising chemical intuition, they struggle with correct and consistent multi-step reasoning. Notably, we find that using prompting strategy and fine-tuning a specialist model on our proposed dataset increases performance by 50% over the leading closed-source model. We hope that oMeBench will serve as a rigorous foundation for advancing AI systems toward genuine chemical reasoning.[1]

1 Introduction

Organic reaction mechanisms, the stepwise elementary reactions where reactants form intermediates and products, provide a detailed, step-by-step description of how chemical bonds are broken and formed during an organic chemical reaction (Muller, 1994). In other words, a mechanism serves as the natural algorithm of a reaction.

[1]https://anonymous.4open.science/r/oMeBench-6E54

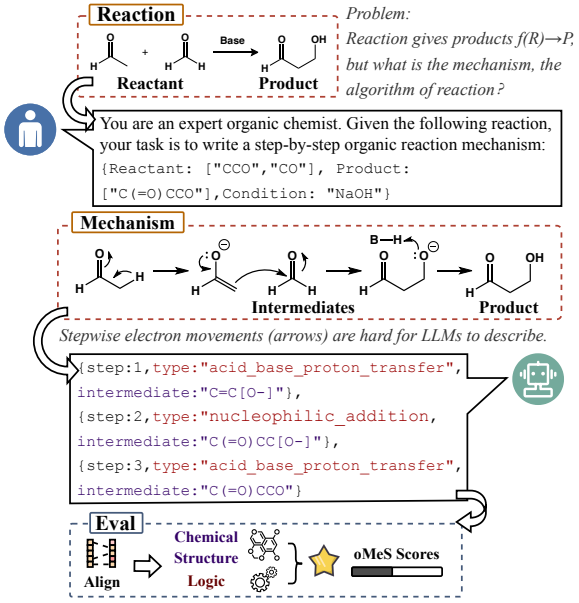


Figure 1: Given a reaction from oMeBench, we generate a reaction mechanism and score it with oMeS.

Mechanisms form the foundation for understanding chemical reactions (Grossman, 2003), and play a key role in accelerating molecular discovery (Carey and Sundberg, 2007; Smith, 2025). For instance, in drug discovery, mechanistic models serve as the basis for predicting metabolic pathways and toxicities (Silverman and Holladay, 2014; Patrick, 2023). Thus, organic reaction mechanisms are not merely theories of fundamental chemistry but tools that drive innovations in applied chemistry.

Large language models (LLMs) have recently demonstrated remarkable performance in diverse chemical applications, such as property prediction (Stark et al., 2022), retrosynthetic analysis (Chen et al., 2023), product prediction (Schwaller et al., 2021), reaction mining (Zhong et al., 2023), action extraction (Zhong et al., 2024), and reaction-condition optimization (Broni-Bediako et al., 2022). Despite this progress, chemical reasoning, especially *mechanistic reasoning*, remains largely unexplored. Existing

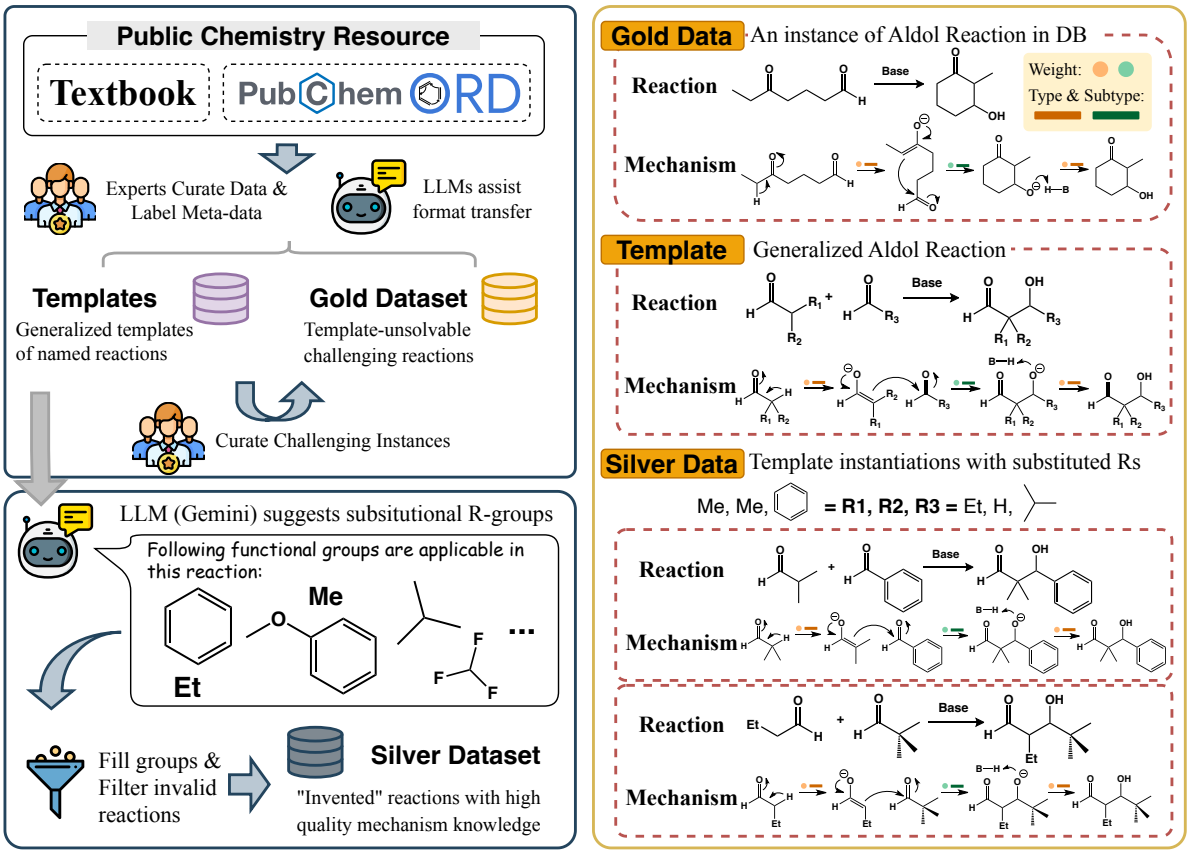


Figure 2: Overview of dataset construction and examples. A named reaction refers to a class of reactions that share a common mechanistic pattern and can be abstracted into a generalized template. Templates are denoted using placeholders such as "R-groups," which represent variable substituents (e.g., Me, Et, H)

benchmarks primarily focus on mapping reactants to products without explicit annotation of the underlying elementary steps (Lowe, 2012; Kearnes et al., 2021), leaving models unable to explain *why* or *how* a reaction occurs. Moreover, prior attempts to construct reaction mechanism datasets largely rely on rule-based systems (Chen et al., 2024), which encode a limited set of hard-coded transformation rules (e.g., deprotection). This results in relatively simple datasets with low reaction diversity, and therefore models which don't generalize well.

To address this gap, we introduce **oMeBench** (Fig. 1), which contains over 10,000 annotated mechanistic steps with intermediates, step-type labels, and rationales, as well as a wide range of reaction classes and difficulty levels. To enable fine-grained evaluation, we further propose **oMeS**, a dynamic scoring framework that aligns predicted and gold-standard mechanisms using weighted similarity metrics. Together, these resources allow us to dissect model behavior beyond product prediction and quantify mechanistic fidelity (mechanism-level accuracy). Our extensive evaluation reveals that while LLMs demonstrate non-trivial chemical intuition, they struggle to sustain multi-step causal logic, especially in complex or lengthy mechanisms. Importantly, exemplar-based in-context learning and supervised fine-tuning both yield substantial improvements (Dong et al., 2023; Zhang et al., 2023), highlighting not only the potential but also the current limitations of LLMs in mechanistic chemistry. We hope that oMeBench will serve as a rigorous foundation for advancing AI systems toward mechanistic chemical reasoning. Our contributions are threefold:

- We construct the first expert-annotated large-scale suite of organic reaction mechanism datasets, with step-by-step mechanisms and natural language rationales, all released in a standardized format that is both human-interpretable and machine-readable.
- We introduce novel dynamic and chemically interpretable evaluation metrics that automatically measure LLMs' mechanism elucidation and reasoning.
- We conduct a systematic study of mechanistic reasoning across multiple LLMs. We reveal and partially bridge critical gaps, achieving a 50% final gain over the proprietary baseline.

Dataset	#Temps	Size(Scaled)	Scaling Method	Level	Mech?	Int?	NL?
Lowe (2012)	86	50k	Expert System	Easy	✗	✗	✗
Chen et al. (2024)	86	31k	Expert System	Easy	✓	✗	✗
(Joung et al., 2024)	86	19.2k	Expert System	Easy	✓	✓	✗
Tavakoli et al. (2023)	–	5k+	–	Medium	✓	✓	✗
Li et al. (2025a)	100	100	Manual Curation	Easy	✓	✗	✓
Wang et al. (2023)	30	30	Manual Curation	Easy	✗	✗	✓
Ours	196	10k	Manual Curation	Mixed	✓	✓	✓

Table 1: Comparison of representative organic reaction datasets by scale, construction method, and annotation type. "#Temps" denotes unique templates; "Mech?", "Int?", and "NL?" indicate the presence of mechanistic details, intermediates, and natural language descriptions, respectively.

Dataset	#Reactions	#Steps	#Types	#Subtypes	Annotation Source
oMe-Gold	196	858	8	30	Expert-verified, curated from textbooks and literature
oMe-Template	167	722	8	30	Expert-curated templates abstracted from oMe-Gold
oMe-Silver	2,508	10,619	8	30	Automatically expanded from oMe-Template with filtering

Table 2: Statistics of the three curated datasets in oMeBench.

2 Related Work

Organic Reaction Datasets. Large-scale reaction datasets, such as USPTO (Lowe, 2012), Pistachio (Mayfield et al., 2017), and PubChem (Kim et al., 2025), collect reactions from the literature without providing explicit mechanistic details. Recent mechanism-aware datasets generally rely on atom-mapped encodings that are difficult to interpret (App. B.2), and remain limited in scope: some focus narrowly on radical (Tavakoli et al., 2023) or polar reactions (Tavakoli et al., 2024), while others are oversimplified with low diversity (Chen et al., 2024; Joung et al., 2024). Broader scientific benchmarks (Li et al., 2025a; Wang et al., 2023) include mechanisms only in simple QA formats.

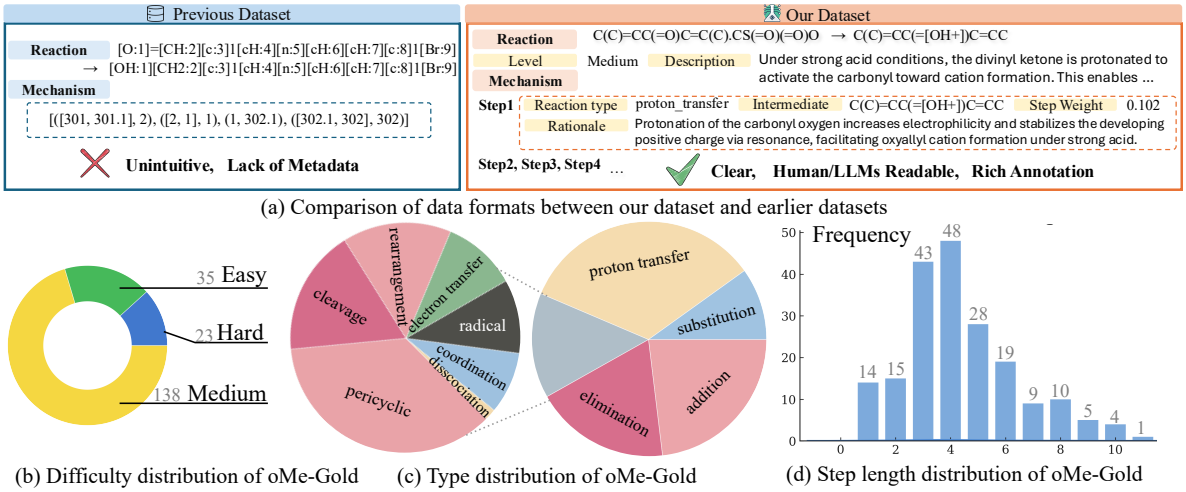
Chemistry-Specific Models. Domain-specific models, such as ChemDFM (Zhao et al., 2024) exploit chemical corpora via pretraining and fine-tuning, while Ether0 (Narayanan et al., 2025) enhances multi-step reasoning through supervised and reinforcement learning. mCLM (Edwards et al., 2025) has advanced this direction by modularization at the representation level, modeling molecules as functional building blocks. Despite these advances, most chemistry-specific LLMs focused on tasks such as molecular property prediction, retrosynthesis, or text-based QA, with explicit mechanistic reasoning still largely unexplored.

Chemistry-related tasks in LLMs. Research on LLMs in chemistry has evolved from early tasks such as molecular name translation and property prediction (Chithrananda et al., 2020; Zheng et al., 2024b) to more complex reasoning applications, including retrosynthesis and automated synthesis protocols (Wang et al., 2025; Vaškevičius and Kapočiūtė-Dzikienė, 2024). Recent work has explored prompting strategies and autonomous agents to extend model capabilities (Bran et al., 2024; Darvish et al., 2024). However, LLM performance on these tasks remains limited. In particular, understanding organic mechanisms, the foundation of all organic transformations, has received little attention, with prior papers addressing only a few simple cases (Bran et al., 2025). To address this gap, we present the first comprehensive study of LLMs on mechanism reasoning and release a large-scale dataset to support fine-tuning and evaluation.

3 Dataset

Tab. 1 highlights three key limitations of early datasets: (i) **unreadable formats**. Each mechanistic step is represented through atom-mapping, which is non-intuitive and difficult to interpret (App. B.2); (ii) **missing metadata**, limiting deeper analysis; and (iii) **lack of difficulty and diversity**, as most reactions are either high-school level, domain-specific, or repetitive despite large scale.

To address these limitations, we introduce three complementary organic reaction mechanism datasets. **oMe-Gold** comprises literature-verified organic reactions with detailed mechanisms, serving as the gold-standard benchmark for evaluation. **oMe-Template** consists of mechanistic templates that generalize families of reactions sharing a common mechanistic pathway. Each template contains substitutable placeholder *R-groups*, enabling instantiation into diverse reaction instances. **oMe-Silver** is a large-scale dataset systematically expanded from **oMe-Template**, con-



(a) Comparison of data formats between our dataset and earlier datasets

(b) Difficulty distribution of oMe-Gold (c) Type distribution of oMe-Gold (d) Step length distribution of oMe-Gold

Figure 3: Overview of the dataset metadata and format.

taining chemically plausible but not necessarily literature-verified reactions, and primarily used for large-scale model training. All three datasets include rich expert annotations covering difficulty. Tab. 2 summarizes the dataset statistics, and Fig. 2 illustrates the complete construction process.

3.1 Gold Dataset

All reactions were curated from authoritative organic chemistry textbooks and reaction databases (Kim et al., 2025; Kearnes et al., 2021; Grossman, 2003; Kurti and Czakó, 2005). For each named reaction,[2] we included either the classical instance or a challenging variant. In addition, several out-of-domain reactions from textbooks were included to enhance diversity. Reactions were initially extracted using Gemini-Pro-2.5 (Team et al., 2023) and subsequently verified by experts for SMILES validity and chemical consistency.[3]

Difficulty Level. Difficulty is defined by mechanistic complexity and the depth of reasoning required. Reactions are classified as **Easy** (20%), involving single-step logic; **Medium** (70%), requiring conditional reasoning; and **Hard** (10%), involving complex mechanisms that demand advanced understanding. All reactions were independently annotated by two expert chemists, with disagreements resolved by a third expert (App. A).

Step Type and Subtype. We propose a two-level taxonomy for elementary steps, comprising 8 types and 30 subtypes, detailed in Tab. 5. The annotation procedure follows *Difficulty Level*.

[2]A named reaction refers to a class of reactions following a shared mechanistic template.

[3]Among the 196 entries, 189 required manual correction. Manual verification was essential, as SOTA LLMs and other tools often fail to produce chemically valid results.

Descriptions and Rationales. Each entry includes a high-level natural language description of the reaction and a rationale for every mechanistic step explaining the underlying mechanistic principles. These texts were generated using Gemini-Pro-2.5 through prompts in App. H.2.

Step Weighting. Each step is weighted by its mechanistic importance and reasoning complexity. We 1) assign initial weights to each step type, 2) adjust for unusually simple or difficult subtypes, and then 3) fine-tune by positional or structural factors. The complete details are in App. C.1.

3.2 Template Abstraction and Expansion

oMe-Template comprises template-level reactions from named reaction databases, which provide generic *R-group* structures and representative instances. As these sources lack explicit mechanistic details, expert chemists reconstructed each mechanism by generalizing from the instance-level reaction while preserving its reaction conditions. This abstraction is chemically justified, since named reactions inherently represent families of reactions sharing common mechanistic pathways. Metadata annotation followed the same protocol as **oMe-Gold**. Representative examples are in App. B.2.

oMe-Silver is constructed by expanding **oMe-Template**. For each template, Gemini-Pro-2.5 proposes chemically plausible R-group substitutions (App. H.1), and invalid SMILES are filtered automatically. Metadata are inherited from the source template, and the scaling algorithm is in App. D. On average, each template yields 15 valid silver-standard reactions. This LLM-guided expansion enables greater chemical diversity and plausibility than rule-based substitution.

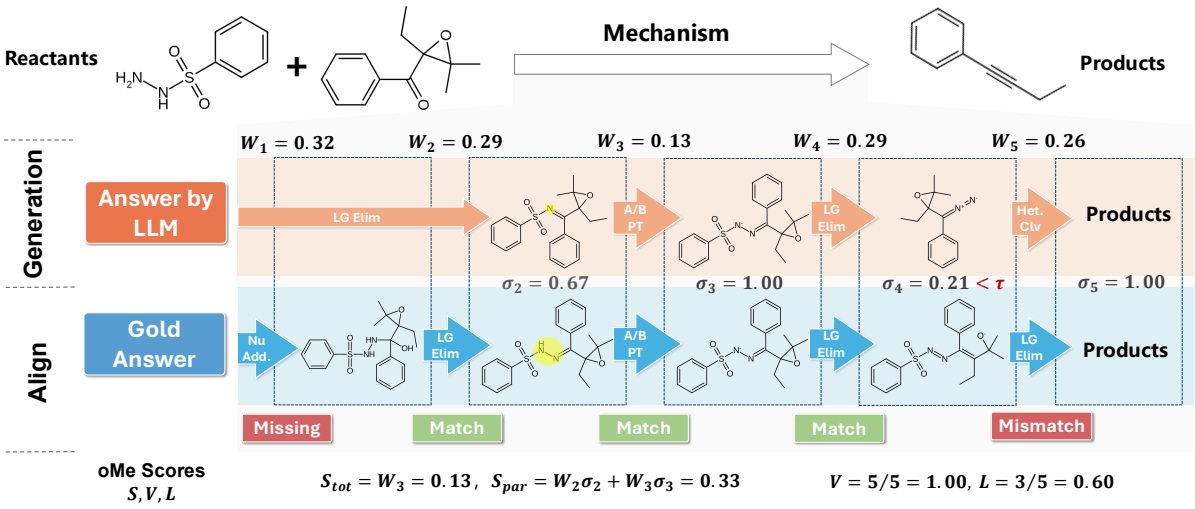


Figure 4: The overview of the evaluation system. LLM-generated mechanisms are dynamically aligned with gold references. Subtype correctness, molecular similarity σ, and weight W are used to compute oMeS S, V, and L.

4 Evaluation

We introduce **oMeS**, a dynamic evaluation framework offering four complementary metrics for fine-grained assessment of LLMs' mechanistic reasoning. It addresses three main challenges: (i) unreliable LLM-as-judge results due to weak domain knowledge, (ii) the inadequacy of static string matching for complex mechanistic outputs, and (iii) the need to score partial reasoning. **oMeS** resolves this through dynamic alignment (§ 4.2) and weighted partial scoring (§ 4.1). The evaluation pipeline (Fig. 4) consists of three stages: (i) Generate: Prompt the LLM to produce stepwise mechanisms, each with an intermediate (SMILES) and selected reaction subtype (Tab. 5). The full prompt is in App. H.3. (ii) Align: Match the predicted and gold mechanisms using a weighted Needleman–Wunsch dynamic algorithm (Likic, 2008). (iii) Calculate: Derive four complementary metrics from the alignment to enable detailed, fine-grained evaluation of mechanistic reasoning.

4.1 Definitions

Let the gold mechanism be $G = \{(t_k, I_k, w_k)\}_{k=1}^{N}$ with N steps, where t_k is the gold subtype label, I_k is the gold intermediate SMILES, and $w_k \geq 0$ is the step weight with $\sum_{k=1}^{N} w_k = 1$. Let the model's raw prediction be $P = \{(\hat{t}_j, \hat{I}_j)\}_{j=1}^{M}$ with M predicted intermediates. After alignment (§ 4.2), each gold step k is matched to at most one predicted step; we denote such an aligned prediction at position k as $(\hat{t}_k, \hat{I}_k)$ or marked missing if unmatched.

SMILES Validity Score (V) measures a model's ability to produce chemically valid intermediates, defined as the fraction of predicted SMILES that can be parsed by RDKit.

Logical Fidelity Score (L) measures stepwise mechanistic labeling accuracy irrespective of structural correctness, computed over aligned positions. Formal definitions of V and L are in App. E.

Organic Mechanism Scores (S_{tot}, S_{part}) measures LLMs' reasoning ability in organic reaction mechanisms. For each step, subtype correctness is a prerequisite, since the model can demonstrate understanding of the underlying chemical rationale only with the correct subtype. **oMeS-total** (S_{tot}) will assign credit only for *exact* matches:

$$s_k^{\text{tot}} = w_k \, \mathbf{1}_{\{t_k = \hat{t}_k\}} \, \mathbf{1}_{\{I_k = \hat{I}_k\}}, \qquad S_{\text{tot}} = \sum_{k=1}^{N} s_k^{\text{tot}}$$

oMeS-partial (S_{part}) grants partial credit when the generated intermediate has meaningful chemical similarity to the gold intermediate, as measured by molecular fingerprints, while **oMeS-total** is stricter. Let $\mathbf{T}$ denotes the *Tanimoto similarity* (Bajusz et al., 2015) of Morgan fingerprints (Rogers and Hahn, 2010) and τ is a similarity threshold:

$$\sigma(I_k, \hat{I}_k) = \mathbf{T}(I_k, \hat{I}_k) \cdot \mathbf{1}_{\{\mathbf{T}(I_k, \hat{I}_k) \geq \tau\}}$$

$$S_{\text{part}} = \sum_{k=1}^{N} s_k^{\text{par}}; \quad s_k^{\text{par}} = w_k \, \sigma(I_k, \hat{I}_k) \, \mathbf{1}_{\{t_k = \hat{t}_k\}}$$

4.2 Alignment Strategy

We align $G = \{(t_k, I_k, w_k)\}_{k=1}^{N}$ and $P = \{(\hat{t}_j, \hat{I}_j)\}_{j=1}^{M}$ with Needleman–Wunsch algorithm:

Model	Size	Overall				Easy				Medium				Hard			
		V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}
Proprietary and Frontier Models																	
GPT-5	-	90.4	51.9	23.6	29.1	92.4	69.9	42.5	51.8	90.0	50.7	21.6	26.5	90.2	31.5	7.2	10.0
GPT-4o	-	83.7	20.3	3.9	5.0	82.6	26.0	8.3	10.1	83.8	20.6	3.2	4.2	85.1	9.7	1.6	1.6
o3	-	90.8	47.4	24.0	28.1	94.6	64.5	36.8	47.3	90.6	45.6	23.3	26.3	86.4	32.5	8.6	9.7
Claude-Sonnet-4	-	91.0	37.3	14.9	17.9	93.0	54.3	28.9	33.9	90.9	35.8	12.7	15.6	88.7	20.7	7.5	7.9
Gemini-Pro-2.5	-	90.5	57.8	34.9	37.9	94.5	70.5	56.5	61.3	90.3	56.6	32.4	35.2	85.4	46.3	16.6	18.5
DeepSeek-R1*	685B	87.6	45.4	22.5	25.0	88.4	62.3	39.0	41.9	87.9	45.1	20.6	23.3	82.9	21.3	8.3	9.7
Open Source Models																	
GPT-OSS	20B	78.2	31.5	12.9	14.2	88.4	47.6	25.1	28.0	77.0	30.3	11.7	12.6	70.2	14.8	1.8	2.4
LLaMA-3	8B	73.5	12.6	2.7	3.3	69.9	13.0	1.5	1.5	74.2	13.2	3.2	4.0	74.9	7.9	1.5	1.8
Qwen-3	4B	58.4	14.9	3.2	4.2	63.1	12.1	2.6	2.6	60.4	17.5	3.8	5.3	40.3	4.3	0.2	0.3
Mistral	7B	52.8	12.7	2.0	2.2	56.6	14.6	1.7	2.2	52.5	12.8	2.3	2.5	48.6	9.0	1.0	1.0
Phi-4	7B	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Domain-Specific Models																	
OLMo-2	13B	17.4	6.1	0.6	0.6	22.0	10.2	3.1	3.1	16.5	5.2	0.0	0.0	16.2	4.9	0.0	0.0
OpenBioLLM	8B	4.4	1.7	0.3	0.3	9.4	1.7	0.3	0.3	3.8	1.9	0.3	0.3	0.0	0.0	0.0	0.0
ChemDFM	8B	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BioT5+	770M	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 3: Model Performance on oMeBench Across Difficulty Levels

- **Match:** if $t_k = \hat{t}_j$, we evaluate both exact-match (s^{tot}) and similarity-based credit (s^{par});
- **Type mismatch:** if $t_k \neq \hat{t}_j$, we assign zero credit and a mild penalty;
- **Gaps (skip):** skipping a gold step or a predicted step incurs a fixed penalty.

To break ties, we prioritize alignment paths sequentially by (1) higher S_{tot}, (2) higher S_{part}, (3) fewer mismatches, and (4) fewer gaps. The final traceback assigns one of four step-level tags: `match`, `type_mismatch`, `skip_gold` (missing prediction), or `extra` (spurious prediction). The alignment procedure is robust to common edge cases, including minor SMILES typos or benign variants, skipped or extra steps, and cases where the intermediate is plausible but the subtype is mislabeled. Case studies and ablations are in App. F.

5 Experiments

5.1 Can LLMs solve mechanism problems?

To establish the baseline performance on our benchmark, we evaluate multiple different LLMs under common one-shot prompting setup. We adopt four metrics: **Validity (V)**, **Logicality (L)**, **oMeS-total** (S_{tot}) and **oMeS-partial** (S_{part}).[4] The results are presented in Tab. 3 and Fig. 5.

Evaluated Models. We categorize the 14 LLMs into three groups: (i) **Proprietary LLMs**: GPT-series (Achiam et al., 2023), o3 (OpenAI, 2025), Claude-sonnet-4 (Anthropic, 2025), and Gemini Pro 2.5 (Team et al., 2023); (ii) **Open-source LLMs**: DeepSeek-R1 (Guo et al., 2025) and instruct version of LLaMa-3 (Dubey et al., 2024), Qwen-3 (Yang et al., 2025), Mistral, and Phi-4-mini (Abdin et al., 2024); (iii) **Domain-specific LLMs**: OLMo-2 (OLMo et al., 2025), OpenBioLLM (Ankit Pal, 2024), ChemDFM (Zhao et al., 2024), and BioT5+ (Pei et al., 2024).

[4]Detailed prompts are in App. H.3.

5.2 LLMs show limited understanding of mechanisms

The evaluation indicates that while current LLMs can often produce syntactically valid SMILES and maintain a certain degree of procedural coherence, their understanding of reaction mechanisms remains highly limited. Even the strongest frontier models achieve only moderate scores that decline sharply with task difficulty. For example, *Gemini-Pro-2.5*, the best-performing model, reach only 37.9 on S_{part}, while many other models yield scores close to zero, reflecting a profound lack of mechanistic knowledge. In other words, LLMs can imitate the surface form of mechanistic explanations but fail to capture the precise, multi-step causal logic underlying organic reactions.

Domain-specific models underperform. Although chemistry-trained models (e.g., ChemDFM, OLMo-2) are optimized on large chemical corpora, their performance lags behind that of general-purpose LLMs. Even OLMo-2, the strongest, underperforms open-source baselines across all metrics, including even validity. This disparity suggests that while domain pretraining exposes models to vast chemical literature, it fails to provide explicit mechanistic reasoning. The frequent generation of irrelevant or inconsistent structure indicates that these models capture surface-level chemical patterns rather than causal mechanistic logic.

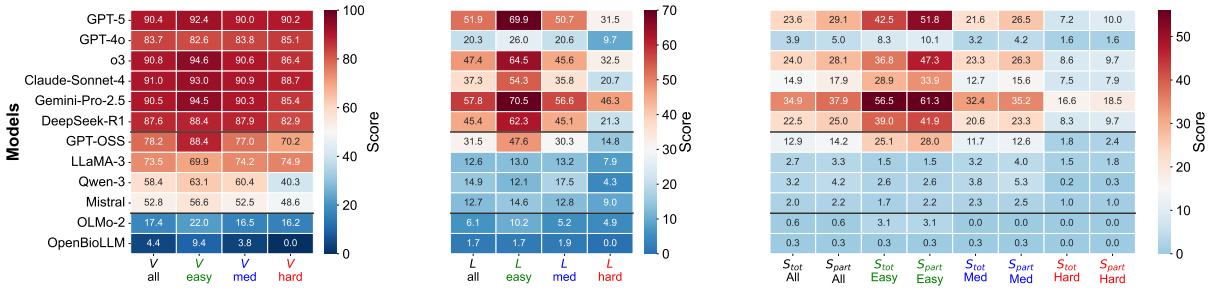


Figure 5: Performance of LLMs on oMeBench across difficulty levels. Frontier models outperform others but all degrade on harder reactions, while chemistry-specific models lag despite domain pretraining

Chain-of-Thought fails to enhance reasoning. One-shot Chain-of-Thought (CoT) prompting (Wei et al., 2022) did not yield measurable improvements (Tab. 14). The generated reasoning traces tend to be verbose yet shallow, merely paraphrasing reaction conditions without offering causal insights. Unlike symbolic or mathematical reasoning, mechanism elucidation requires a structural understanding, which CoT, without chemistry-aware supervision, fails to provide.

5.3 Why do LLMs fail?

To systematically analyze the failure modes, we categorize three broad categories of errors: *Chemical Knowledge Errors*, *Mechanistic and Procedural Errors*, and *Structural and Formalization Errors*. Each of them is further divided into representative subtypes (Tab. 8). These categories are not independent. For instance, misinterpreting reaction conditions can lead to misidentified reactive centers or invalid intermediates. In Fig. 6, we analyze Gemini-2.5-pro as a representative.

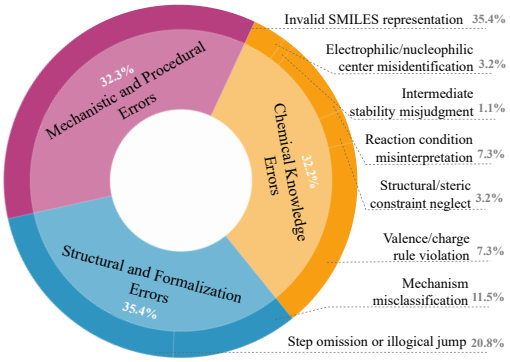


Figure 6: Distribution of mechanistic reasoning errors for Gemini-Pro-2.5. The chart groups errors into three primary categories, with each bar indicating the percentage of a specific subtype.

Chemical structural inconsistency. Despite high product-level SMILES validity in Tab. 3, 35.4% of all errors arise from invalid or inconsistent SMILES within multi-step mechanisms. Models can often produce valid final products but fail to maintain coherent symbolic representations across steps. In addition, valence and charge violations account for 7.3% of total errors, revealing instability in representing intermediate states. The case in Tab. 4 shows how LLMs invent inconsistent chemical structures to provide a logical mechanism, which is considered as a **perceptual gap** of LLMs.

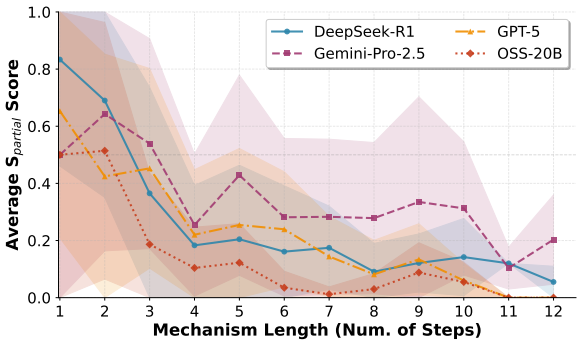


Figure 7: Performance vs Reaction Complexity. Performance degradation as length increases. Longer pathways amplify illogical jumps, showing models' limited ability to sustain causal reasoning across extended steps.

Length effect of the mechanism. Fig. 7 shows that performance consistently drops with increasing mechanism length. While short two- or three-step mechanisms can often be reproduced, mechanisms longer than five steps frequently omit intermediates or rearrangements, leading to incomplete or illogical pathways. This highlights their inability to sustain multi-step causal reasoning over longer contexts. Skipping critical steps, accounting for 20.8% of total errors, reflects the models' difficulty in sustaining multi-step causal reasoning. An example of this behavior is in Tab. 4.

Inherently more challenging types. Mechanism step types differ markedly in difficulty. Fig. 8 shows models perform better on frequent, pattern-based transformations such as substitution and addition, but struggle with rearrangements, pericyclic reactions, and radical processes. These categories demand deeper causal reasoning about electron

Step	Model Output	Comments
1	**acid_base_proton_transfer:** `CC(C1CC1)=N[N-]...`	Correct initial deprotonation.
2	**acid_base_proton_transfer:** `[CH2-]C(C1CC1)=N[N-]...`	Redundant step: removes α-H instead of cleaving.
3 ...	**leaving_group_elimination:** `C=C(C1CC1)N=N`	Fabricate electron flow and charge distribution.

Table 4: Example where the model achieves the correct result via implausible steps. Full results in App. J.4

flow, intermediate stability, and molecular structure, which are capabilities that current LLMs fail to develop. The observed disparity suggests that model predictions rely primarily on surface pattern association rather than mechanistic understanding. These results reveal two key deficiencies: a **reasoning gap**, where models fail to sustain coherent multi-step causal logic, and a **knowledge gap**, where limited mechanistic understanding leads to systematic errors in complex reaction types.

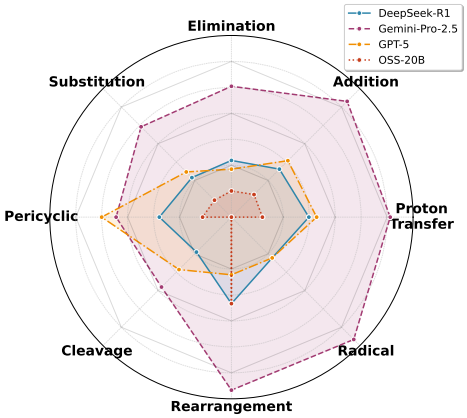


Figure 8: Model accuracy by type. While addition and substitution are handled moderately well, pericyclic, rearrangement, and radical steps remain challenging.

5.4 Improving LLM Performance

In-context learning (ICL). To narrow the **perceptual gap**, we apply a three-shot in-context learning strategy, selecting three reactions from the training corpus **oMe-Silver** with the highest *Tanimoto similarity* of *Differential Reaction Fingerprint* (DRFP) (Probst et al., 2022) to the query reaction. These exemplars provide structural priors that guide the model toward multi-step, causally coherent reasoning. As shown in Tab. 15 and App. J.5, exemplar-based prompting consistently improves scores across nearly all models. The gains are most pronounced for larger proprietary models, such as GPT-5 and Gemini-Pro-2.5. Even models that underperform in the one-shot setting, like GPT-4o, show notable improvements under ICL. Improvements are especially strong for common step types, such as proton transfer and nucleophilic addition, where models can retrieve and adapt familiar mechanistic patterns. This suggests that ICL enhances reasoning not by introducing new knowledge, but by scaffolding the model's retrieval and analogy capabilities. Further, the improvements also demonstrate the high quality of silver standard data.

Fine-tuning. To move beyond exemplar imitation and further narrow the **reasoning** and **knowledge gaps**, we finetune open-source foundation models on our **oMe-Silver** dataset using supervised instruction tuning. All experiments are implemented in the `LLaMA-Factory` framework (Zheng et al., 2024a) and Transformers (Wolf et al., 2020) with BF16 precision on NVIDIA GH200 GPUs. Qwen-3-4B is trained with both standard SFT (data without rationale) and Chain-of-Thought SFT (CoT-SFT, data augmented with natural-language rationales), while LLaMA-3-8B, Mistral-7B, and OLMo-2-13B are optimized using parameter-efficient LoRA. Fine-tuning yields substantial improvements in all dimensions, particularly for in-domain reactions, with Qwen-3-SFT outperforming some proprietary models such as Claude-Sonnet-4 (Tab. 16).

Notably, incorporating CoT reasoning during SFT provides additional gains in mechanistic accuracy under ICL prompting. While ICL enhances reasoning by imitation, fine-tuning internalizes such reasoning patterns as stable model priors. Together, these results demonstrate that explicit mechanistic supervision is the most effective way to transfer mechanism knowledge and reasoning capabilities into open-source LLMs.

6 Conclusion

In conclusion, we present **oMeBench**, a benchmark for evaluating large language models on *organic mechanism reasoning*. Evaluating 14 proprietary and open-source models, we identify persistent gaps in mechanistic understanding, particularly in multi-step causal reasoning and symbolic consistency. To address these challenges, we introduce **oMeS**, a dynamic evaluation framework, and show that both in-context learning and supervised fine-tuning enhance mechanistic reasoning, with our fine-tuned 4B model achieving 50% higher performance than Claude-Sonnet-4. Overall, oMeBench establishes a foundation for advancing chemically grounded reasoning in large language models.

Limitations

We identify key limitations in our models, dataset, and evaluation framework. While several are discussed throughout the paper, we summarize the most critical aspects below for completeness

Models. Both our improved models and existing LLMs still struggle with mechanistic grounding. They sometimes misinterpret reaction conditions or generate chemically invalid intermediates, revealing limited integration between symbolic reasoning and molecular structure constraints. This indicates that current token-level generation lacks explicit enforcement of chemical conservation and contextual awareness.

Dataset. While **oMeBench** includes graduate-level mechanisms across diverse reaction classes, its current scale remains smaller than large outcome-prediction corpora, limiting opportunities for large-scale pretraining. Expanding the dataset to modern reaction classes, such as photoredox, enzymatic, and transition-metal catalysis, would enable broader and more realistic benchmarking. Moreover, while LLM-driven expansion introduces substantial structural diversity, it yields a skewed distribution of substituents; for example, CH3 is disproportionately overrepresented (Fig. 9), which may bias model learning toward common functional groups.

Evaluation. We recruit annotators with strong chemistry expertise (see Appendix A), whose judgments may differ from those of general or non-expert annotators. Although **oMeS** offers a fine-grained and chemically grounded evaluation in both molecular and mechanistic level, it still abstracts molecules into symbolic SMILES representations, which cannot fully capture three-dimensional environments. Incorporating graph- or geometry-based similarity measures could yield a more complete assessment of reasoning accuracy.

Ethical Considerations

Data. All data were obtained from publicly available textbooks, peer-reviewed publications, and open-access reaction databases, in full compliance with relevant legal and ethical standards. No proprietary or confidential sources were used.

Annotation. Human annotations were carried out by qualified chemistry researchers, compensated at the standard institutional rate; anonymized summaries of their expertise and experience are provided in Appendix A.

Model and Environmental Ethics. All models used are open-source or publicly available under research-friendly licenses. Because our methods build upon pretrained language models, they may inherit biases from their underlying training data. Moreover, since most chemical sources are written in English, certain linguistic or regional chemistries may be underrepresented. We also recognize the environmental costs associated with large-scale model computation and therefore restrict our work to evaluation and lightweight fine-tuning.

Responsible Research. We aim to promote fairness, transparency, and reproducibility in AI-for-science research. The tools and benchmarks introduced in this work are designed to assist, not replace, human chemists. All results should be interpreted with expert judgment.

Use Of Generative AI

We employed large language models (e.g., ChatGPT) to assist with grammar refinement, LaTeX code formatting, and, in a few cases, improving the conciseness of lengthy sentences in the authors' original handwritten content. All outputs were subsequently reviewed and critically revised by the human authors to ensure scientific accuracy and originality.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *ArXiv preprint*, abs/2412.08905.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Malaikannan Sankarasubbu Ankit Pal. 2024. Openbiollms: Advancing open-source large language models for healthcare and life sciences. https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B.

Anthropic. 2025. About claude models. https://docs.anthropic.com/en/docs/about-claude/models. Accessed: 2025-09-24.

Dávid Bajusz, Anita Rácz, and Károly Héberger. 2015. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):20.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535.

Andres M Bran, Theo A Neukomm, Daniel P Armstrong, Zlatko Jončev, and Philippe Schwaller. 2025. Chemical reasoning in llms unlocks strategy-aware synthesis planning and reaction mechanism elucidation. *ArXiv preprint*, abs/2503.08537.

Ekow Broni-Bediako, Zhaoyu Li, Lannie Estergreen, José A Vásquez-Cervantes, Gabriel Lee, Mohamad Ali Mashhadi, Alán Aspuru-Guzik, and Klavs F Jensen. 2022. Autonomous reaction optimization with a language model. *Nature Communications*, 13(1):6813.

Francis A. Carey and Richard J. Sundberg. 2007. *Advanced Organic Chemistry: Part A: Structure and Mechanisms*, 5th edition. Springer Science & Business Media, New York, NY.

Shuan Chen, Ramil Babazade, Taewan Kim, Sunkyu Han, and Yousung Jung. 2024. A large-scale reaction dataset of mechanistic pathways of organic reactions. *Scientific Data*, 11(1):863.

Ziqi Chen, Zixun Liu, Zhong-Yu Wei, and Dongxiao Zhao. 2023. G-retro: A graph-based generative model for retrosynthesis prediction. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3637–3645.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2020. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *ArXiv preprint*, abs/2010.09885.

Kourosh Darvish, Marta Skreta, Yuchi Zhao, Naruki Yoshikawa, Sagnik Som, Miroslav Bogdanovic, Yang Cao, Jason E Hein, and Alán Aspuru-Guzik. 2024. ORGANA: A robotic assistant for automated chemistry experimentation and characterization. *ArXiv preprint*, abs/2401.06949.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, and 1 others. 2023. A survey on in-context learning. *ArXiv preprint*, abs/2301.00234.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Carl Edwards, Chi Han, Gawon Lee, Thao Nguyen, Bowen Jin, Chetan Kumar Prasad, Sara Szymkuć, Bartosz A. Grzybowski, Ying Diao, Jiawei Han, Ge Liu, Hao Peng, Martin D. Burke, and Heng Ji. 2025. mclm: A function-infused and synthesis-friendly modular chemical language model. *ArXiv preprint*, abs/2505.12565.

Robert B. Grossman. 2003. *The Art of Writing Reasonable Organic Reaction Mechanisms*, 2nd edition. Springer Science & Business Media, New York, NY.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv preprint*, abs/2501.12948.

Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi S. Jaakkola. 2017. Predicting organic reaction outcomes with weisfeiler-lehman network. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2607–2616.

Joonyoung F Joung, Mun Hong Fong, Jihye Roh, Zhengkai Tu, John Bradshaw, and Connor W Coley. 2024. Reproducing reaction mechanisms with machine-learning models trained on a large-scale mechanistic dataset. *Angewandte Chemie International Edition*, 63(43):e202411296.

Steven M Kearnes, Michael R Maser, Michael Wleklinski, Anton Kast, Abigail G Doyle, Spencer D Dreher, Joel M Hawkins, Klavs F Jensen, and Connor W Coley. 2021. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826.

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. 2025. Pubchem 2025 update. *Nucleic acids research*, 53(D1):D1516–D1525.

Laszlo Kurti and Barbara Czakó. 2005. *Strategic applications of named reactions in organic synthesis*. Elsevier.

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, Eisuke Kawashima, NadineSchneider, Dan Nealschneider, Andrew Dalke, tadhurst cdd, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, Alain Vaucher, and 11 others. 2025. rdkit/rdkit: 2025_03_5 (q1 2025) release.

Hao Li, He Cao, Bin Feng, Yanjun Shao, Xiangru Tang, Zhiyuan Yan, Li Yuan, Yonghong Tian, and Yu Li. 2025a. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *ArXiv preprint*, abs/2505.21318.

Hao Li, He Cao, Bin Feng, Yanjun Shao, Xiangru Tang, Zhiyuan Yan, Li Yuan, Yonghong Tian, and Yu Li. 2025b. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *ArXiv preprint*, abs/2505.21318.

Vladimir Likic. 2008. The needleman-wunsch algorithm for sequence alignment. *Lecture given at the 7th Melbourne Bioinformatics Course, Bi021 Molecular Science and Biotechnology Institute, University of Melbourne*, pages 1–46.

Daniel Mark Lowe. 2012. *Extraction of chemical structures and reactions from the literature*. Ph.D. thesis.

John Mayfield, Daniel Lowe, and Roger Sayle. 2017. Pistachio: Search and faceting of large reaction databases. In *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY*, volume 254. AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA.

Paul Muller. 1994. Glossary of terms used in physical organic chemistry (iupac recommendations 1994). *Pure and applied chemistry*, 66(5):1077–1184.

Siddharth M. Narayanan, James D. Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G. Rodriques, and Andrew D. White. 2025. Training a scientific reasoning model for chemistry. *ArXiv preprint*, abs/2506.17238.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, and 1 others. 2025. 2 olmo 2 furious. *ArXiv preprint*, abs/2501.00656.

OpenAI. 2025. Introducing openai o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`.

Graham L Patrick. 2023. *An introduction to medicinal chemistry*. Oxford university press.

Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. 2024. Biot5+: Towards generalized biological understanding with iupac integration and multi-task tuning. *ArXiv preprint*, abs/2402.17810.

Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. 2022. Reaction classification and yield prediction using the differential reaction fingerprint drfp. *Digital discovery*, 1(2):91–97.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754.

Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. 2021. Predicting the outcomes of organic reactions with a language model. *ACS Central Science*, 7(7):1165–1171.

Richard B Silverman and Mark W Holladay. 2014. *The organic chemistry of drug design and drug action*. Academic press.

Michael B Smith. 2025. *March's advanced organic chemistry: reactions, mechanisms, and structure*. John Wiley & Sons.

Hannes Stark, Octavian-Eugen Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. 2022. Chemberta-2: Towards a better molecular foundation model. *ArXiv preprint*, abs/2209.01712.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Mohammadamin Tavakoli, Yin Ting T Chiu, Pierre Baldi, Ann Marie Carlton, and David Van Vranken. 2023. Rmechdb: A public database of elementary radical reaction steps. *Journal of chemical information and modeling*, 63(4):1114–1123.

Mohammadamin Tavakoli, Ryan J Miller, Mirana Claire Angel, Michael A Pfeiffer, Eugene S Gutman, Aaron D Mood, David Van Vranken, and Pierre Baldi. 2024. Pmechdb: A public database of elementary polar reaction steps. *Journal of Chemical Information and Modeling*, 64(6):1975–1983.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *ArXiv preprint*, abs/2312.11805.

Mantas Vaškevičius and Jurgita Kapočiūtė-Dzikienė. 2024. Language models for predicting organic synthesis procedures.

Haorui Wang, Jeff Guo, Lingkai Kong, Rampi Ramprasad, Philippe Schwaller, Yuanqi Du, and Chao Zhang. 2025. LLM-augmented chemical synthesis and design decision programs. *ArXiv preprint*, abs/2505.07027.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *ArXiv preprint*, abs/2307.10635.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *ArXiv preprint*, abs/2505.09388.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey. *ArXiv preprint*, abs/2308.10792.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, and 1 others. 2024. Chemdfm: a large language foundation model for chemistry. *ArXiv preprint*, abs/2401.14818.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024a. Llamafactory: Unified efficient fine-tuning of 100+ language models. *ArXiv preprint*, abs/2403.13372.

Yizhen Zheng, Jiaxin Ju, Yaw-Ling Koh, Xinyue Wang, Chen Gong Lu, Shenda Poon, Tong Liu, Geoffrey Kishun Chan, and Shirui Pan. 2024b. Large language models for scientific synthesis, inference and explanation. *Nature Machine Intelligence*.

Ming Zhong, Siru Ouyang, Yizhu Jiao, Priyanka Kargupta, Leo Luo, Yanzhen Shen, Bobby Zhou, Xianrui Zhong, Xuan Liu, Hongxiang Li, Jinfeng Xiao, Minhao Jiang, Vivian Hu, Xuan Wang, Heng Ji, Martin Burke, Huimin Zhao, and Jiawei Han. 2023. Reaction miner: An integrated system for chemical reaction extraction from textual data. In *Proc. The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP2023) Demo Track*.

Xianrui Zhong, Yufeng Du, Siru Ouyang, Ming Zhong, Tingfeng Luo, Qirong Ho, Hao Peng, Heng Ji, and Jiawei Han. 2024. Actionie: Action extraction from scientific literature with programming languages. In *Proc. The 62nd Annual Meeting of the Association for Computational Linguistics (ACL2024)*.

A Annotation Detail

Annotator Background. We recruited four expert annotators with strong backgrounds in chemistry competitions. Two are gold medalists of the International Chemistry Olympiad, while the other two are first-prize winners at the provincial level. All annotators have received more than four years of rigorous training in organic chemistry, including functional group transformations, reaction mechanisms, stereochemistry, and common strategies in synthetic design, and are currently pursuing degrees in chemistry. To ensure both competence and reliability, the selection criteria required (i) at least a provincial first prize in chemistry competitions and enrollment in a relevant major, (ii) demonstration of responsibility and enthusiasm during interviews, and (iii) successful completion of pilot annotations that reflected correct understanding of the task.

Annotation Procedure. The annotation process began with the authors curating candidate reactions and defining preliminary criteria for reaction *type*, *subtype*, and *difficulty level*. Annotators were first introduced to the project background and refreshed their knowledge of key organic chemistry concepts before receiving detailed task instructions. They were encouraged to refine and clarify the labeling criteria through iterative discussions with the authors. The core tasks involved (i) extracting reactions and validating intermediate SMILES strings, and (ii) annotating metadata such as difficulty level, type, and subtype. In later stages, annotators also contributed to labeling error categories and drafting representative case studies.

Inter-Annotator Agreement. Each reaction was independently annotated by two domain experts, and disagreements were resolved through consultation with a third annotator followed by consensus discussion. To quantify annotation consistency, we measured inter-annotator agreement across major labeling dimensions. Fleiss' κ for reaction types reached 0.95, indicating almost perfect agreement, while the quadratic-weighted κ for difficulty levels was 0.86, reflecting strong consistency across annotators. These results demonstrate that the annotation protocol yields highly reliable and reproducible human labels, providing a stable foundation for subsequent similarity-based evaluation.

A.1 Instructions to Expert Annotators

Task Definition. Given a **reaction entry** with its `reactants_smiles`, `products_smiles`, and `conditions`, your task is to construct or verify a complete and correct **mechanism**. Each entry must include all required metadata fields, ensuring machine readability and chemical validity. The target JSON schema is as follows:

```
{
 "reaction_id": "XX-xxx",
 "level": "easy/medium/hard",
 "name": "...",
 "reactants_smiles": [...],
 "products_smiles": [...],
 "conditions": "...",
 "mechanism_step_nums": 4,
 "description": "...",
 "mechanism": [...]
}
```

Required Metadata Fields. Annotators must carefully fill out or verify the following components:

1. **reaction_id:** A unique alphanumeric identifier (e.g., "NR-201").
2. **level:** Difficulty level: *easy*, *medium*, or *hard*, based on mechanistic complexity and reasoning depth.
3. **name:** The formal or commonly accepted name of the reaction (e.g., "Nazarov Cyclization Reaction").

4. **reactants_smiles / products_smiles:** Chemically valid SMILES strings representing all reactants and products.

5. **conditions:** Concise expression of reaction conditions, including catalysts, reagents, or solvents (e.g., "H^+ OSO_2Me").

6. **description:** A short natural-language summary of the overall mechanism and rationale.

7. **mechanism:** A list of ordered elementary steps. Each step must contain:

   - `step`: Step number (integer, starting from 1).
   - `type`: The high-level mechanistic category (e.g., *addition*, *elimination*, *proton_transfer*).
   - `subtype`: The specific mechanistic subtype (e.g., *acid_base_proton_transfer*, *nucleophilic_addition*).
   - `intermediate_smiles`: A chemically valid SMILES representing the intermediate after this step.
   - `rationale`: A brief expert-level explanation of the mechanistic principle underlying this transformation (e.g., "Protonation of the carbonyl oxygen increases electrophilicity and stabilizes the developing positive charge via resonance.")

Annotation Principles.

1. Each mechanism must align with those recordings in literature, textbooks or databases.

2. All SMILES strings, including reactants, products and intermediates, must satisfy chemical validity and are **compilable by RDKit**.

3. Assign step types and subtypes according to the standardized taxonomy provided in Tab. 5.

4. Annotators should provide rationales and descriptions in **concise academic language**, emphasizing reactivity, resonance, or stability reasoning rather than descriptive repetition.

Example Annotation.

Mechanism JSON

```
{
  "reaction_id": "NR-201",
  "level": "medium",
  "name": "Nazarov Cyclization Reaction",
  "reactants_smiles": ["C(C)=CC(=O)C=C(C)", "CS(=O)(=O)O"],
  "products_smiles": ["CC1=CC(=O)CC1(C)"],
  "conditions": "H+ OSO2Me",
  "mechanism_step_nums": 4,
  "description": "Under strong acid conditions, the divinyl ketone is protonated to activate the
    ↪ carbonyl toward cation formation. This enables oxyallyl cation generation, which undergoes
    ↪  a Nazarov 4$\pi$-electrocyclization to form a cyclopentenyl cation. Subsequent
    ↪ deprotonation and tautomerization yield the thermodynamically stable cyclopentenone
    ↪ product.",
  "mechanism": [
    {
      "step": 1,
      "type": "proton_transfer",
      "subtype": "acid_base_proton_transfer",
      "intermediate_smiles": "C(C)=CC(=[OH+])C=CC",
      "rationale": "Protonation of the carbonyl oxygen increases electrophilicity and stabilizes
    ↪ the developing positive charge via resonance, facilitating oxyallyl cation formation under
    ↪  strong acid."
    },
    ...
}
```

B Dataset

B.1 Related Work

A wide range of datasets have been developed to support machine learning in chemistry. General-purpose chemical databases such as Pistachio (Mayfield et al., 2017) and PubChem (Kim et al., 2025) provide millions of reactions and molecular entries compiled from ELNs, patents, and scientific publications. While valuable for large-scale data mining, these resources mainly emphasize reaction outcomes or compound properties, without explicit mechanistic details.

Among reaction-focused datasets, the USPTO collections are the most widely used. USPTO-50k (Lowe, 2012) and USPTO-MIT (Jin et al., 2017) provide thousands of reactions extracted from patents, but they primarily encode transformations from reactants to products without mechanistic rationales. Later variants such as mech-USPTO-31k (Chen et al., 2024) and USPTO-Mechanistic (Joung et al., 2024) introduced template-based mechanism generation. However, these datasets are generated by automated matching procedures with atom-mapped SMILES, resulting in limited mechanistic diversity and repetitive patterns. They fail to reflect the stepwise reasoning chemists employ when analyzing electron flow and intermediates.

Other efforts have attempted to collect mechanistic steps more explicitly. RMechDB (Tavakoli et al., 2023) focuses on radical reactions, aggregating over 5,000 manually curated steps linked to specific transition states. Its counterpart, PMechDB, specializes in polar reactions (Tavakoli et al., 2024), providing a complementary collection of elementary steps. While valuable, both databases focus narrowly on single mechanistic families, and do not integrate radical and ionic logic in unified pathways. Along with USPTO family, their step collections remain fragmented, making them less suitable for benchmarking end-to-end reasoning. Moreover, this mapping representation is suitable for training neural networks, e.g. GNN, but far from a good dataset for LLM fine-tuning and evaluation.

Recently, datasets like ChemCoTBench (Li et al., 2025b) and SciBench (Wang et al., 2023) introduced natural language chemistry tasks, including multiple-choice questions or descriptive reasoning. These benchmarks emphasize natural language interaction but lack explicit molecule-level mechanistic detail, limiting their ability to test fine-grained chemical reasoning.

In summary, existing datasets fall into two extremes: large-scale outcome-oriented corpora without mechanisms, or narrowly focused collections that lack breadth and natural language interpretability. Our benchmark complements these efforts by combining stepwise mechanistic detail, validated intermediates, and natural language rationales in a unified, diverse, and challenging dataset.

B.2 Data format and comparison

B.2.1 mech-USPTO-31k Sample

Below we show the data format in mech-USPTO-31k (Chen et al., 2024), corresponding to a single line in the CSV file. We can see that early datasets primarily rely on **atom-mapping** to represent individual mechanistic steps, which is non-intuitive and difficult to interpret. Moreover, in the raw CSV format, it is difficult to determine which mechanistic steps together constitute a complete reaction.

A mechanistic step in mech-USPTO-31k

Original Reaction:

```
O=C(OCc1ccccc1)[NH:1][CH2:2][CH2:3][CH2:4][CH2:5][C@@H:6]([C:7]([O:8][CH3:9])=[O:10])[NH:11][C:12](=[O:13])[NH:14][c
      :15]1[cH:16][c:17]([O:18][CH3:19])[cH:20][c:21]([C:22]([CH3:23])([CH3:24])[CH3:25])[c:26]1[OH:27]>>[NH2:1][CH2:2][
      CH2:3][CH2:4][CH2:5][C@@H:6]([C:7]([O:8][CH3:9])=[O:10])[NH:11][C:12](=[O:13])[NH:14][c:15]1[cH:16][c:17]([O:18][
      CH3:19])[cH:20][c:21]([C:22]([CH3:23])([CH3:24])[CH3:25])[c:26]1[OH:27]
```

Updated Reaction:

```
[O:201]=[C:101]([O:202][CH2:203][c:204]1[cH:205][cH:206][cH:207][cH:208][cH:209]1)[NH:1][CH2:2][CH2:3][CH2:4][CH2:5][C@@H
      :6]([C:7]([O:8][CH3:9])=[O:10])[NH:11][C:12](=[O:13])[NH:14][c:15]1[cH:16][c:17]([O:18][CH3:19])[cH:20][c:21]([C
      :22]([CH3:23])([CH3:24])[CH3:25])[c:26]1[OH:27].[BrH:301]>>[NH2:1][CH2:2][CH2:3][CH2:4][CH2:5][C@@H:6]([C:7]([O:8][
      CH3:9])=[O:10])[NH:11][C:12](=[O:13])[NH:14][c:15]1[cH:16][c:17]([O:18][CH3:19])[cH:20][c:21]([C:22]([CH3:23])([CH3
      :24])[CH3:25])[c:26]1[OH:27]
```

Mechanistic Class: *Cbz_deprotection*
Mechanistic Label:

```
[(201, 301.1), ([301.1, 301], 301), (301, 203), ([203, 202], [202, 101]), ([101, 201], 201), (1, 201.1), ([201.1, 201],
      201), (201, [201, 101]), ([101, 1], 1)]
```

B.2.2 oMe-Gold Sample

Below, we present the data format used in our datasets. It provides clear reaction information, including SMILES of reactants, products, and conditions, as well as a step-by-step mechanism with annotations of type, subtype, and intermediate SMILES. In addition, each step is accompanied by fine-grained rationales, and each reaction is accompanied by a natural language description explaining the underlying reasons for the transformations. The dataset is stored in JSON/JSONL format, enabling both large-scale evaluation and fine-tuning, while remaining interpretable for humans and domain experts.

Mechanism JSON


```
{
  "reaction_id": "NR-201",
  "level": "medium",
  "name": "Nazarov Cyclization Reaction",
  "reactants_smiles": ["C(C)=CC(=O)C=C(C)", "CS(=O)(=O)O"],
  "products_smiles": ["CC1=CC(=O)CC1(C)"],
  "conditions": "H+ OSO2Me",
  "mechanism_step_nums": 4,
  "description": "Under strong acid conditions, the divinyl ketone is protonated to activate the
    ↪ carbonyl toward cation formation. This enables oxyallyl cation generation, which undergoes
    ↪  a Nazarov 4$\pi$-electrocyclization to form a cyclopentenyl cation. Subsequent
    ↪ deprotonation and tautomerization yield the thermodynamically stable cyclopentenone
    ↪ product.",
  "mechanism": [
    {
      "step": 1,
      "type": "proton_transfer",
      "subtype": "acid_base_proton_transfer",
      "intermediate_smiles": "C(C)=CC(=[OH+])C=CC",
      "step_weight": 0.1020,
      "rationale": "Protonation of the carbonyl oxygen increases electrophilicity and stabilizes
    ↪ the developing positive charge via resonance, facilitating oxyallyl cation formation under
    ↪  strong acid."
    },
    {
      "step": 2,
      "type": "pericyclic",
      "subtype": "electrocyclization",
      "intermediate_smiles": "C1(=CC(C([CH+]1)C)C)O",
      "step_weight": 0.5714,
      "rationale": "The oxyallyl cation undergoes a conrotatory 4$\pi$-electrocyclization, favored
    ↪  by orbital symmetry and leading to a cyclopentenyl cation stabilized by adjacent
    ↪ substituents."
    },
    {
```

```
      "step": 3,
      "type": "elimination",
      "subtype": "proton_elimination",
      "intermediate_smiles": "C1(=CC(C(=C1)C)C)O",
      "step_weight": 0.2449,
      "rationale": "Deprotonation relieves positive charge on the cyclopentenyl cation, driven by formation of a more conjugated alkene system."
    },
    {
      "step": 4,
      "type": "proton_transfer",
      "subtype": "acid_base_proton_transfer",
      "intermediate_smiles": "CC1=CC(=O)CC1(C)",
      "step_weight": 0.0816,
      "rationale": "Acid-catalyzed tautomerization converts the enol to the more stable conjugated ketone, favored by carbonyl bond strength and extended $\pi$-conjugation."
    }
  ]
}
```


B.2.3 oMe-Template Sample

Substitutable R-groups are denoted as `[*:i]` for RDKit compatibility.

Mechanism JSON


```
{
  "reaction_id": "NR-201",
  "level": "medium",
  "name": "Nazarov Cyclization Reaction",
  "reactants_smiles": ["C(C([*:1]))=CC(=O)C=C(C([*:2]))", "CS(=O)(=O)O"],
  "products_smiles": ["C([*:2])C1=CC(=O)CC1([*:1])"],
  "conditions": "H+ OSO2Me",
  "mechanism_step_nums": 4,
  "description": "The reaction begins with protonation of the carbonyl oxygen in the divinyl ketone, forming a resonance-stabilized oxyallyl cation. This cation undergoes a 4$\pi$-electrocyclic ring closure to yield a cyclopentenyl cation intermediate, followed by deprotonation and keto-enol tautomerization to produce the cyclopentenone product.",
  "mechanism": [
    {
      "step": 1,
      "type": "proton_transfer",
      "subtype": "acid_base_proton_transfer",
      "intermediate_smiles": "C(C([*:1]))=CC(=[OH+])C=CC([*:2])",
      "step_weight": 0.1020,
      "rationale": "Protonation of the carbonyl oxygen activates the divinyl ketone toward cation formation by stabilizing the developing positive charge."
    },
    {
      "step": 2,
      "type": "pericyclic",
      "subtype": "electrocyclization",
      "intermediate_smiles": "C1(=CC(C([CH+]1)C([*:1]))C([*:2]))O",
      "step_weight": 0.5714,
      "rationale": "The oxyallyl cation undergoes a conrotatory 4$\pi$-electrocyclization, generating a cyclopentenyl cation stabilized by conjugation and substituent effects."
    },
    {
      "step": 3,
      "type": "elimination",
      "subtype": "proton_elimination",
      "intermediate_smiles": "C1(=CC(C(=C1)C([*:2]))C([*:1]))O",
      "step_weight": 0.2449,
      "rationale": "Deprotonation of the cation intermediate restores aromatic conjugation and reduces charge localization."
    },
    {
      "step": 4,
```

```
      "type": "proton_transfer",
      "subtype": "acid_base_proton_transfer",
      "intermediate_smiles": "C([*:2])C1=CC(=O)CC1(C([*:1]))",
      "step_weight": 0.0816,
      "rationale": "Final tautomerization converts the enol to the stable conjugated ketone,
    ↪ completing the Nazarov cyclization."
    }
  ]
}
```

B.3 Types and Subtypes

Type	Subtype	Description
cleavage	heterolytic_cleavage	Bond breaks with both electrons going to one fragment (ions formed), including decarboxylation in some condition
	homolytic_cleavage	Bond breaks evenly with one electron to each fragment (radicals formed)
addition	nucleophilic_addition	Nucleophile attacks an electrophilic center
	electrophilic_addition	Electrophile adds to an electron-rich system (e.g., alkene)
	radical_addition	Radical adds to a π-bond system
elimination	proton_elimination	β-H proton of carbocation is abstracted to form a double bond
	leaving_group_elimination	Leaving group departs simultaneously with double bond formation
	radical_elimination	Radical eliminates a small fragment to form a π-bond
substitution	nucleophilic_substitution	Nucleophile replaces a leaving group
	electrophilic_substitution	Electrophile replaces a hydrogen atom on an aromatic ring
	radical_substitution	Radical replaces a group within a σ-bond
rearrangement	1,2-shift	1,2-alkyl or hydride shift
	radical_rearrangement	Radical migrates to a different position
proton_transfer	acid_base_proton_transfer	Proton migrates between a donor and an acceptor
electron_transfer	single_electron_transfer	Single electron moves from donor to acceptor
coordination	lewis_acid_base_coordination	Lone pair coordinates with an empty orbital
radical	radical_initiation	Generation of radicals (e.g., by light, heat)
	radical_propagation	Radical continuously generates new radicals
	radical_termination	Two radicals combine to terminate chain reactions
	radical_coupling	Two radicals form a new bond directly (e.g., C–C or C–N)
pericyclic	cycloaddition	Two π-systems combine to form a ring (e.g., Diels–Alder)
	electrocyclization	Ring closure or opening via conjugated π-system rotation
	sigmatropic_rearrangement	Migration of σ-bond across a π-system (e.g., [3,3]-Claisen)
	group_transfer	Group is transferred via concerted cyclic pathway
	ene_reaction	Transfer of an allylic hydrogen and formation of a new σ-bond
	cheletropic_reaction	Two new bonds formed to the same atom (subset of cycloaddition)

Table 5: Reaction Types, Subtypes, and Descriptions

C Weighting

C.1 Weighting Strategy

Concretely, each step's raw weight combines: (i) a base weight by step *type* (e.g., pericyclic > rearrangement > elementary proton transfer), (ii) a *subtype* modifier (e.g., sigmatropic or cheletropic steps are up-weighted relative to baseline nucleophilic additions), and (iii) contextual adjustments:

- **Positional adjustment** P_{position}: terminal steps that mainly "copy" product structure are slightly down-weighted;
- **Structural bonus** B_{special}: ring closures, key bond-forming events, etc., receive small positive bonuses.

The raw weight for step i is

$$w_i^{\text{raw}} \;=\; \text{clip}\Big(W_{\text{type}} \times M_{\text{subtype}} \times P_{\text{position}} \;+\; B_{\text{special}},\; 0.5,\; 6.0\Big),$$

and the final weights are normalized: $w_i = w_i^{\text{raw}} \big/ \sum_{k=1}^{N} w_k^{\text{raw}}$. This scheme (curated by three domain experts) mirrors how chemists prioritize steps and ensures stability while preserving contrast between trivial and non-trivial reasoning.

C.2 Weighting Implementation

For reproducibility, we summarize in Table 6 the weighting configuration used in our benchmark. The final step weight is computed by combining these factors as described in Section 4.2 (Metrics).

Factor	**Category**	**Values**	**Notes**
Base type weights	Pericyclic	4.0	Highest complexity (cycloadditions, sigmatropic shifts)
	Rearrangement	3.5	Includes 1,2- and 1,3-shifts
	Radical	3.0	Radical initiation, propagation, coupling
	Electron transfer	3.0	Single- or multi-electron processes
	Substitution	2.5	Nucleophilic/electrophilic substitutions
	Cleavage	2.5	Heterolytic/homolytic bond cleavage
	Addition	2.0	Nucleophilic/electrophilic additions
	Elimination	2.0	Deprotonation, leaving-group elimination
	Coordination	1.5	Metal-ligand binding, weak interactions
	Proton transfer / Dissociation	1.0	Elementary steps, down-weighted
Subtype modifiers	Sigmatropic rearrangement	1.3	Extra difficulty
	Cheletropic reaction	1.3	Extra difficulty
	Cycloaddition / Ene	1.2	Ring closure / pericyclic steps
	Homolytic cleavage	1.2	More complex than heterolytic
	Electrophilic substitution	1.2	Context-dependent selectivity
	Nucleophilic substitution	1.1	
	Leaving-group elimination	1.1	
	Proton transfer (acid-base)	0.9	Simplest step, discounted
Positional modifier	Last step	0.9	Often bookkeeping proton transfers
Special bonuses B_{special}	Ring closure (“cycl" in subtype)	+0.3	
	Key bond-forming step (addition/substitution)	+0.2	Central to overall mechanism

Table 6: Weighting configuration for mechanistic steps.

D Scaling Algorithm

We scale **oMe-Template** into **oMe-Silver** using a generative augmentation procedure as the pseudocode outlined below.

Algorithm 1 Mechanism Template Up-Scaling

Require: Template set **oMe-Template** $\mathcal{T} = \{T_1, \ldots, T_n\}$, LLM $\mathcal{G}$, Validator $\mathcal{V}$ (e.g., RDKit)
Ensure: Valid mechanism dataset **oMe-Silver** $\mathcal{D}_{\text{aug}}$

```
Initialize D_aug ← ∅
for each template T ∈ 𝒯 do
    R ← ExtractRGroups(T)                    ▷ Identify reactive sites / substitutable groups
    S ← QueryLLM(𝒢, T, R)                    ▷ Use LLM to propose candidate R-groups
    C ← EnumerateCombinations(R, S)          ▷ Generate all feasible substitution combinations
    for each combination c ∈ C do
        T_c ← SubstituteRGroups(T, c)        ▷ Construct a new mechanism instance
        if IsValid(𝒱, T_c) then
            D_aug ← D_aug ∪ {T_c}            ▷ Add valid mechanism to dataset
        end if
    end for
end for
return D_aug
```

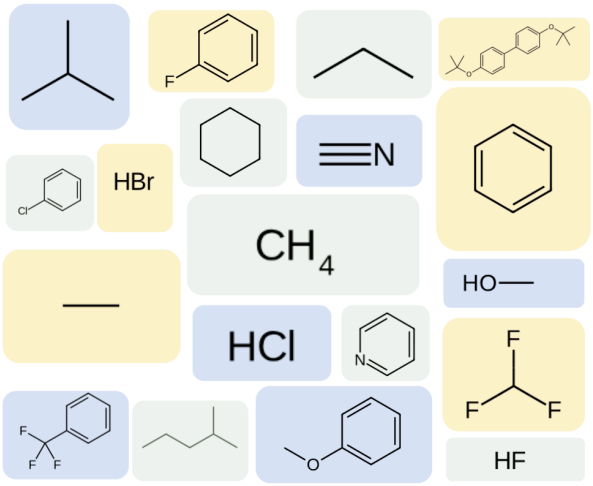


Figure 9: Top 19 R-groups most frequently suggested by Gemini-Pro-2.5

E Metrics Details

The formula for

$$V = \frac{1}{M} \sum_{j=1}^{M} \mathbf{1}\left[\text{RDKit parses } \hat{I}_j\right].$$

:

$$L = \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}\left[t_k = \hat{t}_k\right],$$

where a missing alignment at position k counts as incorrect.

F Alignment Strategy

The alignment strategy is a specialized dynamic programming (DP) algorithm designed to align predicted reaction mechanism steps (**P**) against a set of gold-standard steps (**G**). It significantly improves upon Minimum Edit Distance (MED) and simple match-prioritized scoring by incorporating **chemical structural similarity** and a **hierarchical preference for alignment** into its scoring function.

F.1 Alignment Process and Multi-Criteria Scoring

The alignment strategy is shown as the below,

Pseudocode for Alignment Strategy

Input: - Gold steps $G = [(type, smiles, weight)]_{1..N}$ - Predicted steps $P = [(type, smiles)]_{1..M}$ - Similarity function $sim(\cdot, \cdot)$, threshold τ
Output: oMeSResult $(S_{tot}, S_{part}, Alignment, V, L)$

Initialize DP tables: $DP_tot, DP_par, DP_pen, DP_rank, Trace$
Fill first row/col with skip penalties (skip_gold / skip_pred)
for $i = 1 \rightarrow N$ **do**
  for $j = 1 \rightarrow M$ **do**
    Extract $g = (g_type, g_smi, w)$, $p = (p_type, p_smi)$
    if $g_type = p_type$ **then**
      $m_tot \leftarrow w$ if canonical(g_smi) = canonical(p_smi)
      $m_par \leftarrow w \cdot \sigma(sim(g, p), \tau)$
      action ← “match"
    else
      $m_tot, m_par \leftarrow 0$
      action ← “type_mismatch"
    end if
    Candidates ←
  - Skip_gold: from $(i-1, j)$
  - Skip_pred: from $(i, j-1)$
  - Diagonal: from $(i-1, j-1)$ with (m_tot, m_par)

    Choose best candidate by priority:

$$\max(\ DP_tot, DP_par, Rank, DP_pen\ )$$

    Update DP and Trace at (i, j)
  end for
end for
Backtrace:
Start from (N, M), follow Trace until $(0, 0)$
Build Alignment sequence (match / mismatch / skip)
Metrics:
$S_{tot} \leftarrow DP_tot[N][M]$
$S_{part} \leftarrow DP_par[N][M]$
$V \leftarrow$ fraction of valid predicted SMILES
$L \leftarrow$ fraction of matches in alignment
return $(S_{tot}, S_{part}, Alignment, V, L)$

Unlike traditional sequence alignment, which maximizes a single score or minimizes a single cost, $oMeS$ uses a **four-dimensional, lexicographically ordered score vector** at each DP cell DP$[i][j]$ to determine the optimal path:

$$\text{Key} = (S_{\text{total}}, S_{\text{partial}}, \text{Rank}, S_{\text{penalty}})$$

The optimal action (Match, Type Mismatch, Skip Gold, or Skip Pred) is chosen by maximizing this vector in the order presented (i.e., S_{total} is the primary objective, followed by S_{partial}, etc.). The four fundamental actions used in the DP transitions are described in Table 7.

Action	**Description**
`skip_gold`	A step present in the **Gold** sequence was skipped by the **Pred** sequence.
`skip_pred`	A step present in the **Pred** sequence is extraneous and does not match any **Gold** step.
`match`	The steps align and correctly choose the step types
`type_mismatch`	The steps align but choose the wrong step types.

Table 7: Description of Alignment Actions

- S_{total}: **Strict Match Score (1st Priority):** Rewards perfect alignment where the step **type** is identical and the **canonical SMILES** are identical. This is the highest reward.
- S_{part}: **Partial Chemical Match Score (2nd Priority):** When $S_{\text{total}} = 0$, this metric rewards alignment based on **chemical structural similarity**. It is calculated using the Tanimoto similarity of Morgan fingerprints, but only if the similarity is greater than a threshold $\tau = 0.60$ (Eq. 1).

$$\sigma(s_g, s_p) = \begin{cases} \text{Tanimoto}(s_g, s_p) & \text{if Tanimoto}(s_g, s_p) \geq \tau \\ 0.0 & \text{otherwise} \end{cases} \tag{1}$$

- Rank: **Explicit Operation Priority (3rd Priority):** This ensures the DP prioritizes alignment over skipping when S_{total} and S_{partial} are equal.

  The ranks are defined as:

Action	Rank
match	3
type_mismatch	2
skip_gold / skip_pred	1

- S_{penalty}: Non-Match Penalty (4th Priority): A minor penalty ($-1\text{e}{-6}$) applied to skip and type_mismatch actions. This minimizes the total number of non-perfect alignments when all higher-priority scores are identical.

F.2 Edge Cases Demonstrating Robustness of the Alignment Strategy

We denote each gold mechanistic step as (T_i, I_i), where T_i represents the reaction type and I_i the intermediate. Predicted steps are (T'_j, I'_j). Each alignment result is displayed in two rows: the upper row shows the predicted types T', and the lower row shows the alignment label (*Match*, *Type Mismatch*, *Skip_gold*, *Skip_pred*). Columns labeled "/" indicate redundant predictions that do not correspond to any gold step.

Edge Case 1: Long but redundant predictions. In this situation, the model outputs a long sequence containing extra but chemically valid substeps, such as protonation–deprotonation or intermediate resonance rearrangements. While these steps are redundant, they remain chemically reasonable. Traditional MED or match-first DP tends to greedily match these extra predictions, misplacing true gold correspondences and therefore underestimating the model's performance. Our alignment strategy correctly preserves structural consistency, skipping only redundant steps (`Skip_pred`) without penalizing chemically correct reasoning.

Gold	/	$(T_1{=}\mathrm{A}, I_1{=}S_1)$	/	/	$(T_2{=}\mathrm{B}, I_2{=}S_2)$	$(T_3{=}\mathrm{C}, I_3{=}S_3)$	$(T_4{=}\mathrm{D}, I_4{=}S_4)$
Our Alignment	A Skip_pred	A Match	A Skip_pred	- -	B Match	C Match	D Match
MED / match-first DP	- -	A Match	A Skip_pred	A Skip_pred	B Match	C Match	D Match

In this case, our strategy recognizes that the additional (T_1', I_1') and (T_3', I_3') are redundant but not erroneous, yielding an accurate evaluation of mechanistic reasoning. By contrast, conventional DP misaligns these intermediate steps, lowering both S_{total} and S_{partial} despite chemically valid logic.

Edge Case 2: Incomplete but near-correct predictions. Here, the model predicts a shorter sequence that captures most intermediates correctly but misclassifies the **final step type**. For instance, it correctly generates (C, S_3) and (C, S_5) but predicts (E, S_6) instead of the correct (D, S_6). This single type misclassification leads to a cascade in traditional DP: the alignment path "fetches" the wrong step, propagating mismatches backward and producing multiple `Skip_gold` actions. Our hierarchical scoring isolates this terminal type error without disturbing earlier matches, thus preserving earned partial credit.

Gold	(A, S_1)	(B, S_2)	(C, S_3)	(E, S_4)	(C, S_5)	(D, S_6)
Our Alignment	A Match	C Match	- Skip_gold	- Skip_gold	C Match	E Type Mismatch
MED DP	A Match	C Match	C Match	E Type Mismatch	- Skip_gold	- Skip_gold
Match-first DP	A Match	C Match	C Match	E Type Mismatch	- Skip_gold	- Skip_gold

This example highlights that even a minor terminal error (incorrect type (E, S_6)) can distort traditional DP scoring, leading to overcounted skips and underestimation of reasoning quality. Our approach, however, penalizes only the actual type mismatch while retaining the credit for structurally and sequentially correct steps.

Takeaways. Across both cases, our alignment method:

- Accurately distinguishes between **chemically redundant** and **truly erroneous** steps, preventing underestimation of valid reasoning.
- Localizes **late-step classification errors**, avoiding backward propagation and preserving consistent credit distribution.

ensuring more faithful measurement of multi-step mechanistic reasoning accuracy.

G Extracting High-Similarity Reactions via DRFP

To provide contextually relevant examples for in-context learning or to augment reaction datasets, we develop a retrieval mechanism that identifies structurally similar reactions from an external chemical reaction database. Our approach leverages DRFP to measure similarity between chemical transformations, enabling efficient retrieval of the most analogous reactions.

G.1 Reaction Representation

Given a chemical reaction with reactants $\{R_1, R_2, \ldots, R_m\}$ and products $\{P_1, P_2, \ldots, P_n\}$ represented as SMILES strings, we first construct a unified reaction SMILES representation:

$$\text{RXN}_{\text{SMILES}} = \underbrace{R_1.R_2.\cdots.R_m}_{\text{reactants}} \gg \underbrace{P_1.P_2.\cdots.P_n}_{\text{products}} \tag{2}$$

where "." concatenates multiple molecules and "$\gg$" separates reactants from products.

G.2 DRFP

Traditional molecular fingerprints encode individual molecules as binary or count vectors based on structural features. For reactions, we employ the DRFP approach (Landrum et al., 2025), which captures the structural changes occurring during a chemical transformation.

For each structural feature $k \in \mathcal{K}$ (e.g., circular substructures of varying radii derived from Morgan fingerprints (Rogers and Hahn, 2010)), we compute the feature count in reactants and products:

$$c_k^{(R)} = \sum_{i=1}^{m} f_k(R_i), \tag{3}$$

$$c_k^{(P)} = \sum_{j=1}^{n} f_k(P_j), \tag{4}$$

where $f_k(M)$ denotes the count of feature k in molecule M. The difference fingerprint is then defined as:

$$\mathbf{d} = (d_k)_{k \in \mathcal{K}}, \quad \text{where} \quad d_k = c_k^{(P)} - c_k^{(R)}. \tag{5}$$

A positive value $d_k > 0$ indicates that feature k is generated during the reaction, while $d_k < 0$ indicates consumption. This signed integer vector encodes the net structural changes, making it particularly suitable for comparing reaction mechanisms.

In practice, we use RDKit's `CreateDifferenceFingerprintForReaction` function, which internally hashes the feature space $\mathcal{K}$ into a fixed-dimensional sparse vector via:

$$b = h(k) \bmod N, \tag{6}$$

where $h(\cdot)$ is a hash function and N is the fingerprint dimension. Feature counts are accumulated into corresponding hash buckets, yielding a compact representation.

G.3 Similarity Computation and Retrieval

Given a query reaction with fingerprint $\mathbf{d}_q$ and a candidate reaction with fingerprint $\mathbf{d}_c$ from an external database, we compute their Tanimoto similarity:

$$\text{Sim}(\mathbf{d}_q, \mathbf{d}_c) = \frac{\sum_k \min(|d_{q,k}|, |d_{c,k}|)}{\sum_k \max(|d_{q,k}|, |d_{c,k}|)}, \tag{7}$$

where the absolute values ensure that similarity focuses on the magnitude of structural changes regardless of sign. RDKit's `TanimotoSimilarity` function handles this computation efficiently for sparse count vectors.

Our retrieval algorithm proceeds as follows:

1. **Database preprocessing**: We pre-compute difference fingerprints for all reactions in the external database and cache them as $\{(\mathbf{d}_i, \text{reaction}_i)\}_{i=1}^{|\mathcal{D}|}$.

2. **Query fingerprint generation**: For a query reaction, we compute its difference fingerprint $\mathbf{d}_q$ using the procedure described above.

3. **Similarity scoring**: We compute Tanimoto similarity between $\mathbf{d}_q$ and each cached fingerprint $\mathbf{d}_i$, filtering out identical reactions (by reaction ID or exact SMILES match).

4. **Top-k selection**: We rank candidates by decreasing similarity and retrieve the top k (typically $k = 3$) most similar reactions.

G.4 Implementation Details

We implement this retrieval system using RDKit version 2023.09 with Python 3.9. The external database consists of N_{ext} standardized reactions from diverse sources, stored in JSONL format with fields for reactant SMILES, product SMILES, and reaction metadata (conditions, mechanisms, etc.). The difference fingerprints are computed using Morgan-based circular environments with default RDKit parameters. To avoid computational redundancy, we cache all database fingerprints in memory upon first access.

This fingerprint-based retrieval enables efficient identification of mechanistically similar reactions without requiring explicit reaction classification or rule-based matching, making it well-suited for augmenting language model inputs with relevant chemical knowledge.

H Prompts

H.1 Prompt for suggested group

Prompt Used for R-group Suggestions

You are an expert organic chemist. I need `{num_suggestions}` chemically reasonable substituents for each R-group placeholder.

Reaction Template: `{template}`
R-groups to substitute: `{', '.join(rgroups)}`

Requirements:

1. Provide valid SMILES strings (no placeholder notation)
2. Include diverse functional groups: alkyl, aryl, heteroatoms, etc.
3. Do not include any H (H, CH_3, CH_2CH_3, etc.) in the suggestions, only include C instead (C, CC)
4. Ensure chemical stability and synthetic accessibility
5. Include both electron-donating and electron-withdrawing groups
6. If there are already ring systems in the template, do not include any ring systems in the suggestions

Format your response as JSON, below is an example

```
{
    "[*:1]": ["C", "CC", "C(C)(C)C", "c2ccccc2", "CCCC(C)C",
              "CC(C)(C)OC1=CC=C(C=C1)C2=CC=C(OC(C)(C)C)C=C2",
              "C(F)(F)F"],
    "[*:2]": ["C", "c2ccccc2", "CCCCCCCCCC4=CC2=CC=CC=C2C3=CC=CC=C34",
              "COC1=CC(=CC=C1)C2=CC(=CC=C2O[Si](C)(C)C)C3=CC=CC=C3",
              "OC", "CC(C)(C)C1=CC(OCCC(C)C)=CC=C1"]
}
```

Constraint: Only return the JSON, no other text.

H.2 Prompt used for rationale annoation

Prompt used for rationale annoation

You are an organic chemist. Given a reaction JSON (reactants, products, conditions) and a step-by-step mechanism list without explanations, your task is to:

- Add a top-level `"description"` (overall reasoning, maximum 90 words),
- Add a `"rationale"` (maximum 40 words) to each mechanism step.

You should think like a chemist, and for each step, explain **why** it happens under the given conditions, not just what happens.

Output Format:

```
{
  "description": "overall reasoning",
  "mechanism": [
    {"step": 1, "rationale": "rationale1"},
    {"step": 2, "rationale": "rationale2"},
    ...
  ]
}
```

Hints: In each step's rationale, include at least one *specific chemical reasoning element*, such as:

- Electronic effects: resonance stabilization, inductive effects, conjugation, electron-withdrawing/donating groups.
- Steric effects: bulky substituents blocking attack.
- Reaction conditions: acid/base catalysis, solvent polarity, temperature effects.
- Kinetics vs. thermodynamics: faster pathway vs. more stable product.
- Leaving group ability and stability of intermediates.
- Structure analysis: how the intermediate structure approaches the product.

Key Constraints:

- Refer to placeholders as `R1 ([*:1])`, `R2 ([*:2])`, `R3 ([*:3])`; discuss sterics/electronics qualitatively only.
- **Do not** add, remove, or reorder steps; the number should exactly match the original mechanism list.
- Output **valid JSON only**, and do **not** include any extra text.

Note that the provided hints are designed to follow the logic that human chemists typically adopt when analyzing mechanisms, essentially serving as a chemically grounded "mind map". This design has proven effective in guiding models to produce results with more meaningful underlying chemical analysis.

H.3 Prompt used for mechanism generation

Prompt Used for Mechanism Generation

You are an expert in organic reaction mechanisms.

Task:
Given the following information:**Reactants (SMILES)**,**Products (SMILES)**, **Reaction Conditions**, your task is to write a **step-by-step** organic chemistry reaction mechanism as a **JSON list**.

For each step, include:

1. "`step`" - the step number (starting from 1)
2. "`type`" - the reaction type, selected **exactly** from the predefined list below
3. "`subtype`" - the reaction subtype, selected **exactly** from the predefined list below
4. "`intermediate_smiles`" - a valid SMILES string representing the intermediate at this step (must be parsable by RDKit)

Predefined list of types and subtypes:

```
[
  {"type": "...", "subtype": "...", "description":"..."},
  ... (full content provided in Table 9)
]
```

Output format:
Return only the **valid JSON** in the format below. Do not include any explanatory text, code blocks, markdown formatting, or any other content before or after the JSON.

```
[
  {{ "step": 1, "type": "", "subtype": "", "intermediate_smiles": "" }},
  ...
]
```

Example:

```
Input: # EXAMPLE INPUT OMITTED FOR BREVITY
Expected Output: # EXAMPLE OUTPUT OMITTED FOR BREVITY
```

Input for you:

```
[
  {
    "reactants_smiles": "{ reactants_smiles }",
    "products_smiles": "{ products_smiles }",
    "conditions": "{ conditions }"
  }
]
```

Now, generate your output!

I Error Discussions

Category	Subtype	Manifestation	Cause Analysis
Chemical Knowledge Errors	Reaction condition misinterpretation	Under acidic conditions, the model performs nucleophilic attack or basic deprotonation; under transition-metal catalysis, fails to include coordination or oxidative addition steps.	Models may treat reaction conditions (“acid", “base") as plain text tokens rather than mechanistically decisive entities. Training data often states that a reaction “occurs under acidic conditions" but lacks stepwise mechanistic detail linking conditions to reaction pathways.
	Electrophilic/nucleophilic center misidentification	Nucleophiles attack incorrect atoms or positions; conjugated systems attacked at non-electron-rich sites (e.g., β-position rather than carbonyl carbon).	Deficiency in reasoning about electronic effects (inductive, conjugative, steric). Models may rely only on functional group “canonical" reactivity rather than context-specific micro-environment.
	Valence/charge rule violation	Generated intermediates contain pentavalent carbon, tetravalent nitrogen (non-ammonium), or trivalent oxygen (non-oxonium); misplaced or imbalanced charges.	Represents fundamental violation of chemical rules. As sequence predictors, LLMs lack an internal chemical engine to enforce valence/electron conservation, often producing illogical SMILES patterns.
	Structural/steric constraint neglect	Failure to enforce anti-periplanar geometry in E2/SN2 reactions; impossible pathways in cyclic systems due to steric hindrance.	Stems from difficulty in inferring 3D spatial constraints from 1D/2D textual representations (e.g., SMILES).
	Intermediate stability misjudgment	SN1/E1 mechanisms without carbocation rearrangement; generation of unstable intermediates without subsequent rapid conversion.	Model fails to assess relative energies of intermediates; rearrangements require non-trivial reasoning beyond pattern matching.
Mechanistic and Procedural Errors	figureError	Misclassifies nucleophilic substitution as electrophilic substitution; describes radical reactions using ionic steps; decomposes concerted pericyclic reactions into stepwise ionic events.	Failure to recognize fundamental reaction type (e.g., light-initiated radical vs. ionic pathways).
	Step omission or illogical jumps	Missing key rearrangements; skipping intermediates (e.g., jumping from nucleophilic attack directly to product without deprotonation).	Models may have been exposed to simplified mechanistic descriptions or aim to “optimize" step count, reflecting insufficient grasp of mechanistic rigor.
Structural and Formalization Errors	Invalid SMILES representation	Generated SMILES cannot be parsed (e.g., by RDKit) or parse into chemically nonsensical structures (e.g., open-chain rings).	LLMs struggle with precise symbolic sequences; minor character errors (brackets, ring indices) break validity.
	JSON/field mismatches	Outputs not valid JSON; `type/subtype` contain out-of-vocabulary strings; missing required fields.	Reflects instruction-following limitations. Models may lose adherence to schema constraints over long generations.

Table 8: Taxonomy of error types observed in model-generated mechanisms. Errors are grouped into three broad categories, each with representative subtypes, manifestations, and cause analysis.

J Case Studies

J.1 Case Study 1

Reactants: `C(N)C(=O)c1ccccc1, CCOC(=O)CC(=O)C`
Products: `CCOC(=O)c1c(C)Ncc1c2ccccc2`
Conditions: Acidic medium

Step	Model-Generated Mechanism	Expert Comment
1	**Type**: addition / **Subtype**: nucleophilic_addition **Intermediate**: `CCOC(=O)C(C(=O)C)C(O)(c1ccccc1)CN`	Completely illogical. Both the nucleophilic and electrophilic sites are incorrect, leading to a random adduct that bears no mechanistic connection to the product.
2	**Type**: elimination / **Subtype**: leaving_group_elimination **Intermediate**: `CCOC(=O)C(=C(c1ccccc1)CN)C(=O)C`	Structural errors persist. While the reaction type (elimination) is plausible, the intermediate structure diverges significantly from the correct mechanistic pathway.
3	**Type**: addition / **Subtype**: nucleophilic_addition **Intermediate**: `CCOC(=O)C1=C(c2ccccc2)CNC1(C)O`	Mechanistic reasoning error. The model incorrectly assumes nitrogen participates as a nucleophile again, whereas the reaction should involve the enamine carbon center.
4	**Type**: elimination / **Subtype**: leaving_group_elimination **Intermediate**: `CCOC(=O)C1=C(c2ccccc2)C=NC(C)=C1`	Ring system error. The model inexplicably transforms the five-membered ring into a six-membered one, which is chemically impossible under the given conditions.
5	**Type**: proton_transfer / **Subtype**: acid_base_proton_transfer **Intermediate**: `CCOC(=O)C1=C(C)NC(c2ccccc2)=CC1`	The same ring-size error continues. The overall sequence fails to restore the correct conjugation pattern.
6	**Type**: elimination / **Subtype**: leaving_group_elimination **Intermediate**: `CCOC(=O)c1c(C)N=C(c2ccccc2)C=C1`	Although the final structure matches the product, the mechanistic path remains chemically unsound, indicating weak self-correction ability of the model.

Table 9: Model-generated mechanism for Case Study 1.

Error Analysis. While large language models exhibit clear advantages in handling simple ionic reactions with relatively accurate stepwise reasoning, they remain limited in representing Lewis acid catalysis and complex ring-forming systems. In radical or catalytic cycle reactions, the model struggles to construct a coherent, chemically valid pathway and tends to generate only substrate–product mappings. Moreover, persistent issues include misplaced protonation timing, loss of hydrogen atoms, and poor control of stereochemical details-all of which highlight the current gap between linguistic coherence and genuine chemical reasoning.

J.2 Case Study 2

Reactants: `COc1cc(Br)c(OC)cc1, COC(Cl)Cl, Cl[Ti](Cl)(Cl)Cl, Cl[Ti](Cl)(Cl)Cl`
Products: `COc1cc(Br)c(OC)cc1(C=O)`
Conditions: 1. Lewis acid; 2. Aqueous Hydrolysis

Step	Model-Generated Mechanism	Expert Comment
1	**Type**: heterolytic_cleavage **Intermediate**: `CO[C+]Cl`	Generated Chemically Invalid SMILES. The model doesn't realize that the hydrogen atom should be put into the middle brackets to form a correct SMILES structure. This is a mistake that the model frequently make here and there.
2	**Type**: electrophilic_substitution **Intermediate**: `COc1c(OC)c(Br)c(C(Cl)OC)c[cH+]1`	Incorrect Structure. The model doesn't realize the fact that the electrophilic substitution broke the aromaticity of the ring, thus resulting in the lowercase spelling of benzene ring to be incorrect.
3	**Type**: acid_base_proton_transfer **Intermediate**: `COc1c(OC)c(Br)c(C(Cl)OC)cc1`	Partly Incorrect Step Logic. The proton transfer after an electrophilic addition in an aromatic ring is more often regarded as proton elimination.
4	**Type**: heterolytic_cleavage **Intermediate**: `COc1c(OC)c(Br)c(C=[O+]C)cc1`	Correct
5	**Type**: nucleophilic_addition **Intermediate**: `COc1c(OC)c(Br)c(C(OC)[OH2+])cc1`	Correct
6	**Type**: acid_base_proton_transfer **Intermediate**: `COc1c(OC)c(Br)c(C(O)OC)cc1`	Correct
7	**Type**: acid_base_proton_transfer **Intermediate**: `COc1c(OC)c(Br)c(C(O)[O+](C)H)cc1`	Generated Chemically Invalid SMILES. The order of carbon and hydrogen atoms are incorrect.
8	**Type**: leaving_group_elimination **Intermediate**: `COc1c(OC)c(Br)c(C=[OH+])cc1`	Correct
9	**Type**: acid_base_proton_transfer **Intermediate**: `COc1c(OC)c(Br)c(C=O)cc1`	Correct

Table 10: Model-generated mechanism for Case Study 2.

J.3 Case Study 3

Reactants: `Oc1ccccc1, CC#N`
Products: `CC(=O)c1ccccc1O`
Conditions: i. $ZnCl_2$, HCl; ii. H_2O

Step	Model-Generated Mechanism	Expert Comment
1	**Type**: acid_base_proton_transfer **Intermediate**: `CC##[NH+]`	Incorrect Catalysis. The reaction should be initiated by Lewis acid activation, but the model incorrectly uses a proton as the catalyst. This is a recurring error where the model often ignores the role of Lewis acids.
2	**Type**: electrophilic_substitution **Intermediate**: `Oc1c(C(=[NH+])C)C[cH+]cc1`	Incorrect Structure. The position of the double bond is wrong. The model fails to capture that the hydroxyl group donates electron density through conjugation, which makes the aromatic ring nucleophilic.
3	**Type**: acid_base_proton_transfer **Intermediate**: `CC(=[NH+])c1ccccc1O`	Incorrect SMILES. In addition to misrepresenting the Lewis acid as a proton, the SMILES structure is invalid as it places an unpaired electron on nitrogen, which is chemically unreasonable.
4	**Type**: nucleophilic_addition **Intermediate**: `CC(O[H2+])c1ccccc1O`	Incorrect SMILES. The nitrogen atom is again assigned an unpaired electron, reflecting a repeated formatting error.
5	**Type**: acid_base_proton_transfer **Intermediate**: `CC(O)c1ccccc1O`	Incorrect SMILES. The same issue persists with the misrepresentation of charged nitrogen species.
6	**Type**: acid_base_proton_transfer **Intermediate**: `C(O)c1ccccc1O`	Incorrect SMILES. Once again, the model incorrectly places an unpaired electron on nitrogen.
7	**Type**: acid_base_proton_transfer **Intermediate**: `CC(=[OH+])c1ccccc1O`	Correct.
8	**Type**: acid_base_proton_transfer **Intermediate**: `CC(C=O)c1ccccc1O`	Correct.

Table 11: Model-generated mechanism for Case Study 3.

J.4 Case Study 4

Reaction Overview. Reactants: `CC(C1CC1)=NNS(=O)(=O)c2ccc(C)cc2`
Products: `C=CC1CC1`
Conditions: Base

Step	Model-Generated Mechanism	Expert Comment
1	**Type:** acid_base_proton_transfer **Intermediate:** `CC(C1CC1)=N[N-]S(=O)(=O)c1ccc(C)cc1`	Correct deprotonation of the hydrazone nitrogen. This step aligns with the gold mechanism.
2	**Type:** acid_base_proton_transfer **Intermediate:** `[CH2-]C(C1CC1)=N[N-]S(=O)(=O)c1ccc(C)cc1`	Incorrect second deprotonation: removes an α-hydrogen instead of initiating N–N bond cleavage. Mechanistically implausible.
3	**Type:** leaving_group_elimination **Intermediate:** `C=C(C1CC1)N=N`	Partial match. Correctly identifies diazo formation but misses the ionic character (`[N+]=[N-]`). Electron flow and charge distribution are wrong.
4	**Type:** leaving_group_elimination **Intermediate:** `C=[C-]C1CC1`	Incorrect elimination: the carbocation intermediate is replaced by a carbanion. This reverses the reaction polarity.
5	**Type:** acid_base_proton_transfer **Intermediate:** `C=CC1CC1`	Produces the correct product but through a nonphysical sequence of steps. The mechanism lacks heterolytic cleavage and proper diazonium formation.

Table 12: Model-generated mechanism for Case Study 4.

Expert Analysis In the gold-standard mechanism, the key step is the heterolytic cleavage of the diazonium intermediate to form a carbocation, followed by base-promoted proton elimination. The model, however, omits the cleavage and instead chains multiple proton-transfer and elimination operations. This yields the correct alkene but through a chemically implausible pathway, illustrating that the model captures surface-level reactivity patterns without understanding charge redistribution or intermediate stability.

J.5 ICL Case Study

Reactants: `Nc1ccc(C(=O)C(=O)c2ccccc2)cc1`
Products: `Nc1ccc(C(O)(C(=O)O)c2ccccc2)cc1`
Conditions: NaOH, H_2O

Step	Without ICL	With ICL (Aligned with Gold)
1	**Type:** nucleophilic_addition **Intermediate:** `Nc1ccc(C(=O)C([O-])(O)c2ccccc2)cc1`	**Type:** nucleophilic_addition **Intermediate:** `Nc1ccc(C(=O)C([O-])(O)c2ccccc2)cc1`
2	**Type:** 1,2-shift **Intermediate:** `Nc1ccc(C(O)(C(=O)[O-])c2ccccc2)cc1`	**Type:** 1,2-shift **Intermediate:** `O=C(O)C([O-])(c1ccccc1)c1ccc(N)cc1`
3	**Type:** acid_base_proton_transfer **Intermediate:** `Nc1ccc(C(O)(C(=O)O)c2ccccc2)cc1`	**Type:** acid_base_proton_transfer **Intermediate:** `Nc1ccc(C(O)(C(=O)[O-])c2ccccc2)cc1`
4	– (missing)	**Type:** acid_base_proton_transfer **Intermediate:** `Nc1ccc(C(O)(C(=O)O)c2ccccc2)cc1`

Table 13: A case exemplifies how ICL improves (PT denotes proton transfer). Without ICL, the model omits the critical rearrangement step and terminates early after three steps.

Expert Analysis. Without ICL, the model halts prematurely and fails to include the second proton-transfer step, leaving the mechanism incomplete. With ICL, the model recovers the correct rearrangement pathway and produces all four steps, mirroring the gold-standard mechanism. This example demonstrates that exemplar-guided prompting enhances causal continuity and encourages models to reconstruct multi-step reasoning chains consistent with chemical logic.

K Results

This section presents the performance outcomes of various large language models on **oMeBench** under different reasoning and training configurations. We evaluate models across multiple experimental settings, including *chain-of-thought prompting* (CoT), *in-context learning* (ICL), and *fine-tuning*, to examine how these paradigms influence mechanistic reasoning ability. Each subsection reports quantitative metrics covering validity, linguistic correctness, and both total and partial structural alignment, organized by difficulty level or data domain.

K.1 CoT results

This subsection summarizes model performance under the one-shot *chain-of-thought* setting, where models are prompted to explicitly reason through each mechanistic step before predicting the next intermediate or transformation.

Model	Size	Overall				Easy				Medium				Hard			
		V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}
Open Source Models (One-shot CoT Setting)																	
Mistral	7B	55.2	10.1	3.0	3.5	53.8	20.8	5.1	5.5	56.0	7.3	2.7	3.3	53.0	10.4	1.3	1.9
Qwen-3	4B	49.9	12.6	2.6	3.2	54.3	14.7	1.5	2.6	49.0	13.2	3.1	3.8	48.9	6.0	0.9	1.1
OLMo-2	13B	18.7	6.9	1.1	1.1	13.5	9.2	2.9	2.9	21.1	6.6	0.8	0.9	12.1	5.8	0.0	0.0

Table 14: Performance of Open-Source Models that work under One-shot Chain-of-Thought (CoT) Setting across Difficulty Levels.

K.2 In-context learning (ICL) results

This subsection reports results from the *in-context learning* setting, where models are provided with several exemplars of annotated reaction mechanisms before generating predictions for new queries.

Model	Size	Overall				Easy				Medium				Hard			
		V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}
Hypothetical and SOTA Frontier Models																	
GPT-5	-	94.6 +4.2	66.3 +14.4	46.9 +23.3	50.6 +21.5	93.1 +0.7	75.0 +5.1	56.4 +13.9	61.9 +10.1	94.9 +4.9	66.4 +15.7	45.9 +24.3	49.3 +22.8	95.3 +5.1	52.3 +20.8	38.7 +31.5	41.5 +31.5
GPT-4o	-	78.3 -5.4	44.0 +23.7	23.2 +19.3	26.0 +21.0	81.7 -0.9	45.4 +19.4	26.9 +18.6	32.7 +22.6	77.6 -6.2	43.9 +23.3	23.2 +20.0	25.5 +21.3	77.4 -7.7	41.9 +32.2	17.9 +16.3	18.8 +17.2
o3	-	92.9 +2.1	66.6 +19.2	45.8 +21.8	48.8 +20.7	95.6 +1.0	72.9 +8.4	58.1 +21.3	62.0 +14.7	93.1 +2.5	65.7 +20.1	43.7 +20.4	47.0 +20.7	87.7 +1.3	62.6 +30.1	39.6 +31.0	39.8 +30.1
Claude-Sonnet-4	-	92.6 +1.6	60.2 +22.9	40.2 +25.3	42.2 +24.3	95.4 +2.4	72.5 +18.2	46.9 +18.0	52.9 +19.0	92.3 +1.4	57.9 +22.1	40.7 +28.0	41.8 +26.2	90.7 +2.0	55.2 +34.5	26.3 +18.8	28.7 +20.8
Gemini-Pro-2.5	-	95.2 +4.7	73.8 +16.0	54.4 +19.5	56.9 +19.0	97.5 +3.0	79.9 +9.4	65.7 +9.2	69.4 +8.1	94.9 +4.6	74.8 +18.2	54.6 +22.2	56.8 +21.6	93.1 +7.7	58.4 +12.1	35.5 +18.9	38.4 +19.9
DeepSeek-R1	685B	90.6 +3.0	68.7 +23.3	45.6 +23.1	48.2 +23.2	95.7 +7.3	83.4 +21.1	56.0 +17.0	61.5 +19.6	90.0 +2.1	67.5 +22.4	45.8 +25.2	48.0 +24.7	86.9 +4.0	53.6 +32.3	29.0 +20.7	29.2 +19.5
Open Source Models																	
GPT-OSS	20B	79.3 +1.1	43.1 +11.6	23.1 +10.2	24.0 +9.8	79.0 -9.4	55.6 +8.0	34.7 +9.6	34.9 +6.9	80.0 +3.0	42.9 +12.6	22.6 +10.9	23.6 +11.0	75.6 +5.4	25.0 +10.2	8.6 +6.8	9.6 +7.2
LLaMA-3	8B	77.2 +3.7	44.5 +31.9	19.0 +16.3	20.8 +17.5	77.2 +7.3	40.1 +27.1	16.4 +14.9	20.0 +18.5	76.4 +2.2	46.0 +32.8	20.3 +17.1	21.8 +17.8	73.7 -1.2	41.7 +33.8	15.0 +13.5	15.9 +14.1
Qwen-3	4B	75.6 +17.2	50.1 +35.2	24.6 +21.4	27.0 +22.8	78.8 +15.7	53.1 +41.0	30.5 +27.9	34.3 +31.7	75.7 +15.3	49.8 +32.3	24.9 +21.1	27.0 +21.7	67.0 +26.7	47.2 +42.9	14.0 +13.8	15.7 +15.4
Mistral	7B	63.7 +10.9	38.9 +26.2	16.2 +14.2	17.9 +15.7	68.5 +11.9	42.4 +27.8	20.1 +18.4	23.4 +21.2	63.0 +10.5	39.6 +26.8	16.3 +14.0	17.6 +15.1	64.0 +15.4	29.6 +20.6	10.0 +9.0	11.2 +10.2
Chemistry-specified Models																	
OLMo-2	13B	53.5 +36.1	31.1 +25.0	10.9 +10.3	12.3 +11.7	54.9 +32.9	33.8 +23.6	12.5 +9.4	14.6 +11.5	53.9 +37.4	32.1 +26.9	10.9 +10.9	12.1 +12.1	48.6 +32.4	21.0 +16.1	8.6 +8.6	9.7 +9.7

Score Metrics: ***V*** *(Validity),* ***L*** *(Linguistic Correctness),* S_{tot} *(Total Structural Match),* S_{part} *(Partial Structural Match). (Assumed definitions)*
The second line in each cell shows the change (+/-) from the baseline performance.

Table 15: Model Performance (ICL) with Change from Baseline

K.3 Fine-tuning results

This subsection presents results from *fine-tuned* variants of selected models, comparing their pre- and post-tuning performance under both simple and in-context prompting conditions.

Model	Size	Overall				In-Domain				Out-of-Domain			
		V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}	V	L	S_{tot}	S_{part}
Before Fine-tuning (Simple Prompt)													
Qwen-3	4B	58.5	15.0	3.2	4.3	59.8	16.2	3.0	4.3	54.7	11.4	3.8	4.1
LLaMA-3	8B	73.5	12.6	2.7	3.3	76.9	12.7	2.6	2.9	63.6	12.0	2.9	4.3
Mistral	7B	52.8	12.7	2.0	2.2	54.0	13.0	2.1	2.3	49.0	11.9	1.8	2.1
OLMo-2	13B	17.4	6.1	0.6	0.6	17.4	6.1	0.6	0.6	17.4	6.1	0.6	0.6
After Fine-tuning (Simple Prompt)													
Qwen-3-SFT	4B	85.2 +26.7	38.7 +23.7	18.2 +15.0	20.6 +16.3	86.8 +27.0	43.2 +27.0	20.9 +17.9	23.5 +19.2	80.1 +25.4	24.7 +13.3	9.7 +5.9	11.5 +7.4
Qwen-3-CoT-SFT	4B	82.6 +24.1	37.5 +22.5	15.5 +12.3	17.4 +13.1	83.3 +23.5	42.0 +25.8	18.9 +15.9	20.6 +16.3	80.6 +25.9	24.1 +12.7	5.3 +1.5	7.8 +3.7
LLaMA-3-LoRA	8B	65.0 -8.5	19.6 +7.0	3.9 +1.2	4.8 +1.5	64.7 -12.2	21.7 +9.0	4.4 +1.8	5.3 +2.4	66.0 +2.4	13.5 +1.5	2.3 -0.6	3.4 -0.9
Mistral-LoRA	7B	77.8 +25.0	19.4 +6.7	6.5 +4.5	7.6 +5.4	79.3 +25.3	21.4 +8.4	7.9 +5.8	9.2 +6.9	73.5 +24.5	13.3 +1.4	2.2 +0.4	2.7 +0.6
OLMo-2-LoRA	13B	75.7 +58.3	17.8 +11.7	6.3 +5.7	7.4 +6.8	76.9 +59.5	18.7 +12.6	7.4 +6.8	8.3 +7.7	72.0 +54.6	15.0 +8.9	2.7 +2.1	4.7 +4.1
Before Fine-tuning (ICL Baseline)													
Qwen-3	4B	75.2	50.1	24.6	27.0	78.4	60.3	30.8	32.9	65.8	19.4	6.0	9.2
LLaMA-3	8B	76.2	44.5	19.0	20.8	78.1	52.6	23.7	25.6	70.5	20.2	4.8	6.3
Mistral	7B	64.1	38.9	16.2	17.9	67.6	48.1	20.7	22.9	53.7	11.3	2.9	2.9
OLMo-2	13B	53.5	31.1	10.9	12.3	53.5	31.1	10.9	12.3	53.5	31.1	10.9	12.3
After Fine-tuning (with ICL)													
Qwen-3-SFT	4B	84.4 +9.2	53.6 +3.5	27.4 +2.8	30.6 +3.6	84.7 +6.3	63.3 +3.0	33.8 +3.0	36.5 +3.6	83.7 +17.9	24.4 +5.0	8.1 +2.1	12.9 +3.7
Qwen-3-CoT-SFT	4B	85.6 +10.4	55.1 +5.0	29.2 +4.6	31.0 +4.0	86.5 +8.1	65.5 +5.2	36.3 +5.5	38.0 +5.1	83.0 +17.2	23.8 +4.4	7.8 +1.8	9.9 +0.7
LLaMA-3-LoRA	8B	83.2 +7.0	51.5 +7.0	26.8 +7.8	28.6 +7.8	86.7 +8.6	61.7 +9.1	33.2 +9.5	35.5 +9.9	73.0 +2.5	21.2 +1.0	7.7 +2.9	7.9 +1.6
Mistral-LoRA	7B	82.5 +18.4	45.0 +6.1	22.2 +6.0	24.8 +6.9	84.5 +16.9	55.1 +7.0	28.5 +7.8	31.9 +9.0	76.4 +22.7	15.0 +3.7	3.2 +0.3	3.4 +0.5
OLMo-2-LoRA	13B	79.7 +26.2	39.3 +8.2	21.3 +10.4	22.8 +10.5	82.4 +28.9	46.8 +15.7	25.6 +14.7	27.4 +15.1	71.6 +18.1	16.8 -14.3	8.2 -2.7	8.9 -3.4

*Score Metrics: **V** (Validity), **L** (Linguistic Correctness), S_{tot} (Total Structural Match), S_{part} (Partial Structural Match).*

Table 16: Performance of fine-tuned models under one-shot simple prompt and three-shot ICL prompt

L Fine-Tuning Configuration and Hyperparameters

We fine-tune large language models (LLMs) for chemical reaction mechanism prediction using two complementary parameter-efficient strategies: full supervised fine-tuning (SFT) and Low-Rank Adaptation (LoRA).

L.1 Dataset and Preprocessing

Both configurations utilize the `alpaca_chemical_mechanism_dataset`, formatted in the Alpaca instruction-following template (Taori et al., 2023) with three fields: *instruction* (task description), *input* (reactants and products in SMILES format), and *output* (predicted reaction mechanism).

To accommodate the structured nature of chemical reactions and their detailed mechanistic descriptions, we set a maximum sequence length of $L_{\text{max}} = 6000$ tokens, substantially longer than typical instruction-following tasks. We employ 16 parallel workers for data preprocessing to efficiently handle the large-scale chemical dataset.

L.2 Optimization Hyperparameters

Table 17 summarizes the key hyperparameters for both fine-tuning strategies.

Table 17: Fine-tuning hyperparameters for full SFT (Qwen3-4B) and LoRA (> 4B).

Hyperparameter	**Full SFT**	**LoRA**
Optimization		
Learning rate (η)	1.0×10^{-5}	1.0×10^{-4}
LR scheduler	Cosine	Cosine
Warmup ratio	0.1	0.1
Number of epochs	6	8
Batch Configuration		
Per-device batch size	1	1
Gradient accumulation steps	16	8
Effective batch size	16	8
LoRA-specific		
LoRA rank (r)	—	8
LoRA target modules	—	all
System		
Precision	BF16	BF16
DeepSpeed stage	ZeRO-3	ZeRO-3
Max sequence length	6000	6000

Learning Rate and Schedule: We adopt substantially different learning rates for the two strategies. For full SFT, we use a conservative learning rate of $\eta = 1 \times 10^{-5}$ to prevent catastrophic forgetting of the model's pre-trained chemical and general knowledge. In contrast, LoRA fine-tuning employs $\eta = 1 \times 10^{-4}$, an order of magnitude larger, as only the low-rank adaptation matrices are updated and thus require stronger gradient signals.

Both configurations use a cosine learning rate schedule with a warmup ratio of 0.1, where the learning rate linearly increases from 0 to η during the first 10% of training steps, then follows a cosine decay to 0. This schedule stabilizes early training and prevents aggressive weight updates near convergence.

Distributed Training and Mixed Precision Both configurations leverage DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) for memory-efficient distributed training.