

Grounding Structure in Language: Enriching Symbolic Representations for Molecular Reasoning

Zhengyuan Liu^{†,‡}, Dan He^{‡*}, Nancy F. Chen^{†*}, Wenya Wang^{‡*}

[†]Agency for Science, Technology and Research (A*STAR), Singapore

[‡]Nanyang Technological University, Singapore

{liu_zhengyuan, nancy_chen}@a-star.edu.sg

{dan.he, wangwy}@ntu.edu.sg

Abstract

Large language models (LLMs) show promise for scientific applications but struggle when domain knowledge is encoded in complex structures. Chemistry exemplifies this challenge: molecular structures encode rich relational data that dictates chemical meaning, yet standard formats like SMILES obscure this through arbitrary linearization. In this work, we systematically characterize which structural information lost during linearization most severely hampers LLM reasoning, decomposing the problem into IUPAC-grounded dimensions: substituent positions (locants), attachment points (anchoring), and modification constraints (valency). Our diagnostic reveals non-uniform failures across dimensions, with locant identification being especially challenging. We directly address the failures by grounding the linearized representations in structural context with each component targeting its corresponding diagnostic dimension. General-purpose LLMs with explicit grounding can outperform chemistry-specialized models on structure-aware molecule manipulation, suggesting structural context can compensate for domain-specific training. Moreover, we investigate a training-free inductive protocol that nearly doubles the accuracy of weaker models. These results point toward explicit structural grounding as a principled path to more reliable molecular reasoning across model scales.

1 Introduction

Large language models (LLMs) increasingly assist scientific research (M. Bran et al., 2024), yet they falter when reasoning requires understanding structured representations that differ fundamentally from natural language (Zhang et al., 2025). In science, the structure is not incidental; it *is* the meaning. When processed sequentially by an LLM, the essential connectivity, spatial relationships, and

compositional rules that domain experts rely on for reasoning are fundamentally lost.

Chemistry offers a clean instantiation of this challenge: molecular structures encode rich relational information including atom positions, connectivity patterns, and bonding constraints, which determine chemical meaning (Liao et al., 2024). However, LLMs primarily encounter molecules as SMILES (Simplified Molecular Input Line Entry System) (Weininger, 1988) or SELFIES (SELF-referencing Embedded Strings) (Krenn et al., 2020), both of which are linearized representations and encode the structural relationships implicitly (Leon et al., 2024; Liu et al., 2025).

The consequences are concrete: given a SMILES string like `'C1Oc2cc(O)ccc2C3c4ccccc4OC13'` with a structurally specific instruction like “*add a hydroxyl group at C-3*”, an LLM cannot determine which atom corresponds to “C-3”. The target atom might be the third or other character in the string, depending on how the graph was traversed (Figure 1). Without explicit spatial mapping, the model struggles to execute structural modifications that require positional precision.

This creates what we term the grounding problem: a fundamental disconnect between abstract symbolic representations (text) for LLMs and the structural reality that dictates chemical meanings. The structural information that linearization discards is precisely what International Union of Pure and Applied Chemistry (IUPAC) conventions encode and what chemists rely on for reasoning: locants specifying substituent positions within molecular scaffolds, anchoring points identifying atoms where functional groups attach, and valency constraints governing chemically valid modifications (Ganeeva et al., 2025). While prior work has documented that LLMs struggle with molecular reasoning (Jang et al., 2025a,b), it remains unclear which structural properties underlie these failures and whether they manifest as distinct failure modes.

*Corresponding Authors

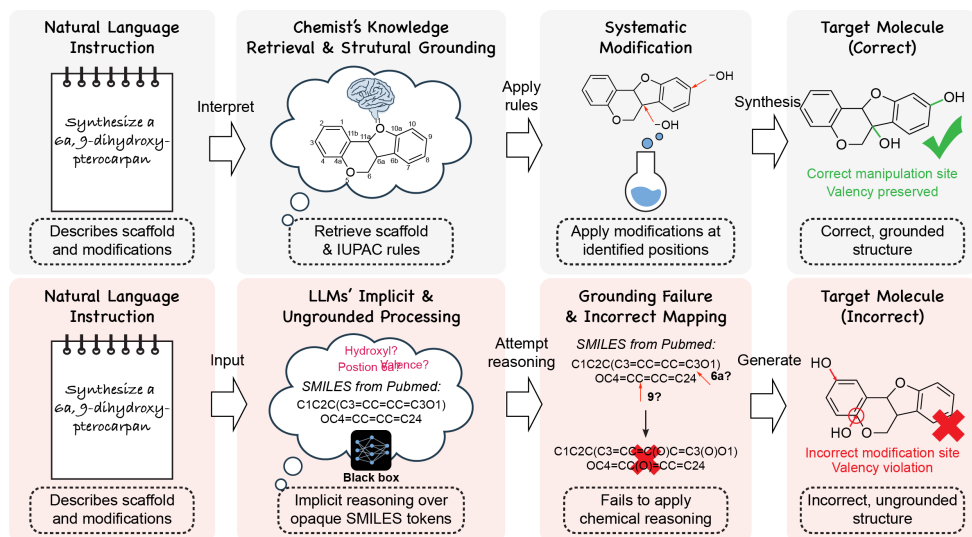


Figure 1: The grounding problem illustrated. Top: Chemists retrieve structural knowledge to correctly interpret and modify molecules. Bottom: LLMs process SMILES as opaque tokens, failing on molecule reasoning.

In this work, we systematically characterize this gap by probing LLMs on these IUPAC-grounded, fundamental structural interpretation capabilities: identifying positional indices, recognizing anchoring atoms, and inferring local bonding environments. Our diagnostic evaluation across different LLMs reveals a universal but non-uniform performance gap on all three dimensions: locant identification is the most challenging for all models, whereas valency reasoning shows strong model-capacity dependence. These findings reveal fundamentally different failure modes across structural interpretation dimensions.

We next propose to restore the discarded structural information explicitly along the three diagnostic dimensions for LLMs to reason with: we enrich the molecular representations with locant-mapped atom indices, anchoring point annotations, and valency and neighbor information derived directly from molecular graphs. Rather than requiring models to implicitly reconstruct molecular structure from string patterns, each augmentation directly addresses a challenge identified in our diagnostic, forming a diagnosis-to-intervention loop. This transforms abstract structural reasoning into pattern matching and constraint satisfaction tasks that are better suited to language model architectures. We evaluate on a dataset built from a subset of ChEBI-20 filtered for caption-to-molecule examples. Our fully enriched representations achieve nearly 70% exact match accuracy with frontier general LLMs, outperforming domain-specialized models. Notably, locant-mapped atom indices yield the largest

gain, consistent with our diagnostic findings.

To further improve the weaker models that struggle to leverage the enriched representations effectively, we develop a teacher-guided reasoning protocol: a stronger model analyzes the failure patterns of weaker models and generates procedural scaffolding for guidance. This training-free approach nearly doubles exact match accuracy, showing the performance gap can be bridged across model scales without any parameter updates. Our contributions are as follows:

1. **Grounding problem characterization.** We decompose the grounding problem into three IUPAC-grounded dimensions: locant identification, anchoring recognition, and valency awareness, and reveal that LLM failures are not uniform: locant identification remains challenging even for frontier models, whereas valency reasoning is strongly model-capacity dependent, suggesting fundamentally different failure modes across dimensions.

2. **Structure-grounded representation enrichment.** We introduce a principled structure-grounded enrichment approach where each component directly targeting the corresponding diagnostic failure and is validated by error reduction analyses. This leads frontier general-purpose LLMs to achieve nearly 70% exact match accuracy on caption-to-molecule generation, surpassing domain-specialized models. Ablation studies confirm that locant mapping yields the largest gain, consistent with our diagnostic findings.

3. **Teacher-guided reasoning protocol.** We leverage a teacher-guided reasoning protocol where

a stronger model analyzes weaker model failures and generates procedural scaffolding, nearly doubling exact-match accuracy without additional training, showing inductive instruction synthesis can bridge performance gaps across model scales.

2 Related Work

Encoding molecular graphs into discrete symbolic sequences is foundational for applying language models to chemistry. Standard linearizations like SMILES (Weininger, 1988) traverse the molecular graph via depth-first search, enabling compatibility but discarding explicit spatial context. While alternatives like SELFIES (Krenn et al., 2020) introduced grammatical constraints to ensure validity, they prioritize syntactic robustness over structural interpretability and lack mechanisms to track atomic positions. Neural modeling wise, previous work explored adding molecular fingerprints to SMILES or integrated graph-based training objectives to better capture topology (Edwards et al., 2022; Christofidellis et al., 2023). Beyond text, multimodal methods have fused language representations with auxiliary encoders, such as graph neural networks in MolCA (Liu et al., 2023) or 2D diagrams and 3D visual renderings in 3D-MoLM (Li et al., 2024), Chem3DLLM (Jiang et al., 2025), and ChemDFM-X (Zhao et al., 2024a). However, these multimodal approaches face a bottleneck: reasoning and generation still occur through textual tokens, requiring the model to implicitly “translate” encoded structural features back into sequential operations without direct grounding.

LLMs continue to struggle with scientific reasoning, particularly tasks requiring spatial and geometric grounding (Jablonka et al., 2024; Boiko et al., 2023). While tool-augmented methods like ChemCrow (M. Bran et al., 2024) can offload precise calculations to external tools, they do not resolve the fundamental disconnect in how models process and interpret molecular strings internally. As a representative benchmark, caption-to-molecule generation evaluates both chemical understanding and structure inference (Edwards et al., 2022; Guo et al., 2023). To better align different modalities, Text2Mol (Edwards et al., 2021) introduced cross-modal retrieval between textual descriptions and molecular graphs, and MolT5 (Edwards et al., 2022) is directly trained on bidirectional translation between SMILES and natural language captions. In more recent works, large language models are

optimized by using massive domain-specific data (Zhao et al., 2024b; Christofidellis et al., 2023) and reinforcement learning (Zhao et al., 2025; Wang et al., 2025; Narayanan et al., 2025).

Despite these advances, systematic evaluation of SMILES reasoning capabilities remains limited. Most benchmarks emphasize aggregate generation quality (validity, uniqueness) rather than specific competencies like positional grounding. Models often succeed through pattern matching on training distribution rather than structural understanding (Guo et al., 2023), echoing findings from Jablonka et al. (2024) that chemistry LLMs frequently generate plausible-seeming but structurally incorrect outputs. Our diagnostic tasks isolate specific reasoning failures, providing mechanistic insight into these observed limitations.

3 Diagnostic Evaluations of Reasoning on Molecule Representations

3.1 Diagnostic Framework

We probe LLMs on three structural interpretation tasks aligned with the IUPAC-grounded dimensions: locant position identification, anchoring point recognition and valency and neighbor labeling. Each task isolates a specific reasoning capability, deliberately decoupling symbol-level relational reasoning from end-to-end molecule generation to directly assess these decomposed skills (Table 1).

Locant Position Identification Given a molecule’s IUPAC name and its SMILES string, and an atom-mapped template where all positions are masked with placeholder indices, a tested model is asked to identify the atom that corresponds to locant ID 1 according to IUPAC nomenclature rules and replace its placeholder with :1. Correctness is evaluated against ground-truth locant assignments generated using RDkit.¹ This task requires models to bridge IUPAC positional conventions and atom symbol ordering of SMILES.

Anchoring Point Recognition Given a substituent name (e.g., “acetyl,” “3-pyridyl”) and an atom-mapped SMILES of each molecule (arbitrary numerical indices were assigned to atoms, see Appendix A), the model must identify the atom index serving as the attachment point. To preclude pattern memorization, atom indices are randomly shuffled across instances in our assessment.

¹<https://github.com/rdkit/rdkit>

Valency and Neighbor Labeling Given an atom-mapped SMILES and a target atom, the model must identify: (1) its neighboring atoms, (2) bond orders of bonds between itself and each neighbor, (3) hydrogen count, and (4) total valency. This probes the capability of parsing bond symbols, tracking ring closures, inferring aromaticity, and applying hydrogen-filling conventions.

3.2 Experimental Setups

Testset Construction We generated a diagnostic testset comprising 600 total samples. By construction, all three tasks are dataset-agnostic and scalable. The samples are primarily sourced from ChEBI-20 (Edwards et al., 2022), with additional diverse substituents drawn from PubChem (Kim et al., 2025) following deduplication. For locant identification, we sampled from ChEBI-20 and selected molecules in various locant numbering conventions including fused polycycles with angular locants (4a, 8a), and bridged systems following path-length conventions. For anchoring recognition, we built a diverse set of substituents from PubChem, spanning carbon (-alkyl, -aryl, -acyl), nitrogen, oxygen, and sulfur attachments, including positional isomers (2-pyridyl vs. 3-pyridyl) requiring disambiguation. For valency and neighbor labeling, we sampled from ChEBI-20, filtered out trivial cases (e.g., oxygen with one neighbor, carbon with two neighbors and standard valence 4) and emphasized atoms with context-dependent valency, like nitrogen in amines (valence 3) and sulfur in sulfides (valence 2).

Evaluation Protocol For location position identification and anchoring point recognition, the predicted index is compared with the ground truth label, and we report accuracy as the result. For valency labeling, we report hierarchical metrics: neighbor set (including neighbor numbers and each neighbor’s atom ID), bond order accuracy, hydrogen count accuracy, total valence accuracy, and exact match of all fields, as shown in Table 1.

Tested Models We evaluate Meta-Llama-3.1-8B (Dubey et al., 2024), Gemma-3-12B (Team et al., 2025), Qwen3-30B-A3B-2507 (Yang et al., 2025), GPT-4o-1120 (Hurst et al., 2024), GPT-5.2 (OpenAI, 2025), Claude-Sonnet-4.5 (Anthropic, 2025), and a state-of-the-art domain-specialized model ChemDFM-v2.0-14B (Zhao et al., 2025). Each model is prompted with task-specific instructions specifying the input format, task requirements, and

A. Locant Position Identification Example

IUPAC name: (3-chlorophenyl)acetic acid
 Input SMILES: O=C(O)Cc1cccc(Cl)c1
 Input Template: O=[C:X](O)[CH2:X][c:X]1[cH:X][c:X](Cl)[cH:X][cH:X][cH:X]1
Expected Output: O=[C:X](O)[CH2:1][c:X]1[cH:X][c:X](Cl)[cH:X][cH:X][cH:X]1

B. Anchoring Point Recognition Example

Substituent: (1R)-1-phenylethyl
 Input SMILES: c1ccccc([CH]C)c1
 Atom-mapped SMILES: [cH:1]1[cH:2][cH:4][cH:8][c:3]([CH:5][CH3:7])[cH:6]1
Expected Answer Atom ID: 5

C. Valence and Neighbor Labeling Example

Input Molecule: n1cccn1
 Atom-mapped SMILES: [*:5]1[cH:1][o:2][cH:3][n:4]1
 The [*:5] is a Nitrogen atom.
Expected Output (from a JSON format):
 Target atom ID: 5, Atom type: Nitrogen
 Hydrogen count: 0, Total valence: 3
 Neighbors and bonds:
 Neighbor Atom ID 1, Atom type: Carbon, Bond type: Aromatic, Bond order: 1.5
 Neighbor Atom ID 4, Atom type: Nitrogen, Bond type: Aromatic, Bond order: 1.5

Table 1: Examples for the diagnostic evaluation tasks. Here we clean the JSON format for better readability. See more design considerations in Appendix A.

expected output structure, accompanied by three few-shot examples randomly sampled from a held-out set with no scaffold overlap with test instances (see Appendix I for full instructions and model generation examples).

3.3 Diagnostic Analysis

Figure 2 presents diagnostic results across seven LLMs on the three structural interpretation tasks, and reveals a clear difficulty hierarchy. Locant position identification is the most challenging: except Claude-Sonnet-4.5, all models achieve below 35% accuracy, with most hovering near the random baseline (17%) even with chain-of-thought (CoT) reasoning (Wei et al., 2022). Anchoring point recognition shows moderate performance (30-50% accuracy vs. 15% random baseline, excluding Claude-Sonnet-4.5 that achieves around 60%), suggesting that attachment connectivity is partially recoverable despite being implicit in SMILES syntax. Valency awareness exhibits the highest but most variable performance: Claude-Sonnet-4.5 and GPT-5.2 achieve nearly 80% accuracy while smaller models like Llama-3.1-8B and Gemma-3-12B fall below 10%. Notably, ChemDFM-2-14B, a chemistry-specialized model, shows no consistent advantage over general-purpose LLMs, indicating

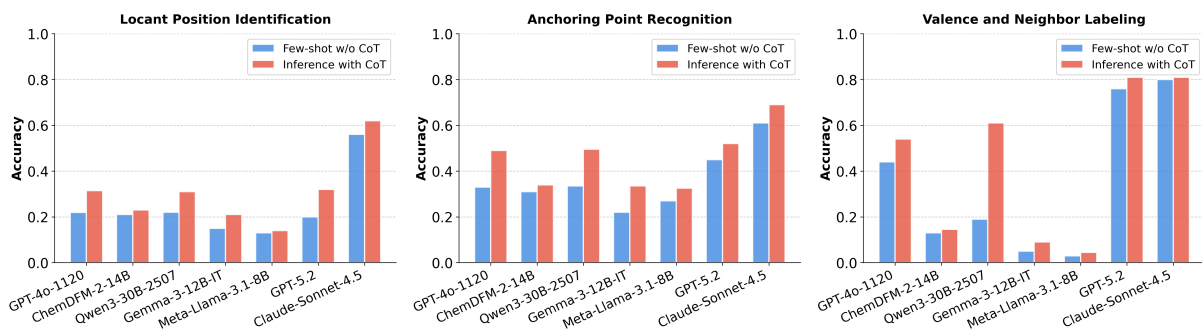


Figure 2: Evaluation results on the three diagnostic tasks. For the valence and neighbor-labeling evaluation, a prediction is counted as correct only if all sub-tests are answered correctly.

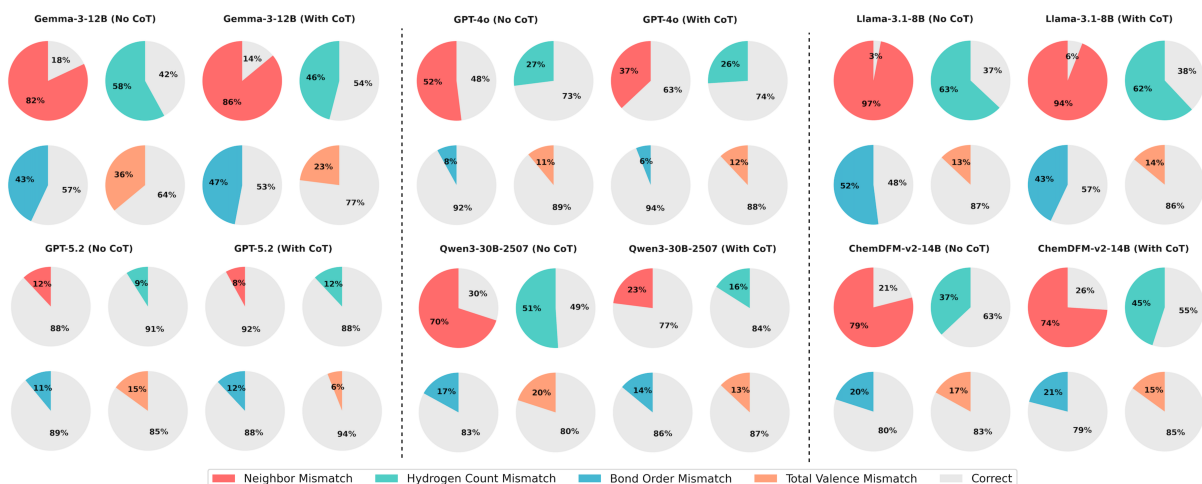


Figure 3: Error type analysis on the valence and neighbor labeling evaluation.

that domain-specific training alone does not resolve the structural grounding gap.

A granular analysis of the valency task demonstrates that models fail in systematically different ways (as in Figure 3).² Neighbor mismatch, the failure to correctly identify atom connectivity, dominates weaker models, accounting for 84% of errors in Gemma-3-12B and 70% in Qwen3-30B. Since identifying connectivity does not require chemical knowledge, this finding tells that the errors stem from structural reasoning gaps: models cannot reliably parse local molecular graphs from linear strings, causing cascading errors in valency assessment. Hydrogen count mismatch is also prevalent, highlighting the models’ inability to accurately infer implicit hydrogens omitted by SMILES. CoT reasoning substantially reduces error rates for Qwen3-30B (e.g., 70% $\rightarrow$ 23% in neighbor mismatch, 51% $\rightarrow$ 16% in hydrogen count mismatch), indicating that explicit reasoning can partially compensate for structural grounding gaps but cannot

eliminate them entirely.

Taken together, these diagnostic results show that LLM failures on structural interpretation are not uniform but reflect fundamentally different underlying mechanisms. Locant identification remains consistently challenging across all models, suggesting that positional mapping represents an inherent limitation of linearized representations rather than a model capacity deficit. Valency reasoning, by contrast, shows strong model-capacity dependence, suggesting failures trace back to structural reasoning gaps rather than missing chemical knowledge. Explicit grounding, rather than implicit inference from linearized representations, is therefore necessary for reliable molecular reasoning across model scales.

4 Enriching Symbolic Representations

We therefore propose to enrich SMILES with the three grounding dimensions identified above. This transforms implicit graph reconstruction (Dwivedi et al., 2021) into explicit sequence inference that are better suited to language model architectures.

²Due to space limit, see Appendix D for the valence and neighbor labeling error analysis of Claude-Sonnet-4.5.

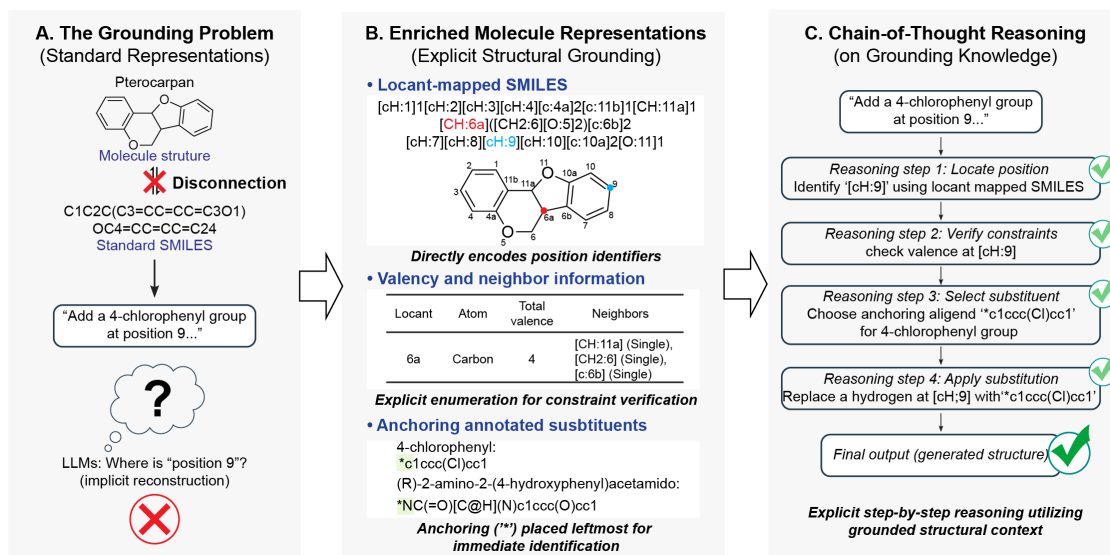


Figure 4: Overview of the enriched symbolic representations for molecule reasoning.

4.1 Representation Components

IUPAC-Aligned SMILES with Locant Tags Position identifiers are embedded into SMILES strings using locant-mapped notation [atom:locant]. For example, molecule ‘D-galactose’ as shown below, [C@H:2] and [CH2:6] directly encode that these carbons are position 2 and 6 respectively according to IUPAC nomenclature rules. This allows a model to process a structurally specific instruction like “*substitution at position 2*,” by locating the corresponding token directly, without reconstructing the numbering system from graph topology.

D-galactose: O=C[C@H](O)[C@@H](O)[C@@H](O)[C@H](O)CO->O=[CH:1][C@H:2](O)[C@@H:3](O)[C@@H:4](O)[C@H:5](O)[CH2:6]O

Locant Valence and Neighbor Information For each labeled position, we explicitly enumerate neighboring atoms and bonding capacity. Each locant entry specifies the atom type, position index, total valence, and neighbors with their bond types. For instance, position 2 in ‘D-galactose’ is annotated with valence 4 and neighbors [CH:1], O, and [CH:3]. This allows models to perform constraint verification on whether a proposed modification respects valence limits or not without parsing implicit hydrogen counts from SMILES syntax where hydrogen atoms are omitted in the standard format.

Anchor-Aligned Substituents Substituent fragments are represented with dummy atoms (*) to mark the attachment positions, and with the anchoring atom placed leftmost in the reordered SMILES

strings. Two examples are shown below. This left-alignment makes attachment points immediately identifiable at the token level, eliminating the need to infer connection sites from chemical knowledge.

N-acetyl- α -D-galactosaminyl:
*[C@H]1O[C@H](CO)[C@H](O)[C@H](O)[C@H]1NC(C)=O

α -L-fucosyl:
*[C@H]1O[C@H](C)[C@H](O)[C@H](O)[C@H]1O

4.2 Enriched Context Construction

Given a natural language description (e.g., caption-to-molecule generation), we extract the named entities of core molecular scaffolds and substituents using GPT-4o (see details in Appendix C). The extracted names are then used to retrieve their molecular information from PubChem (Kim et al., 2025) and OPSIN,³ and their enriched representations were generated using RDKit. As illustrated in Figure 4 and Appendix Figure 12, each scaffold entry contains IUPAC-aligned locant tags (see detailed implementation in Appendix B) and valence and neighbor annotations; each substituent entry contains anchor-aligned SMILES with dummy atom (*) marking the attachment position. The enriched context is provided to the model along with the downstream task instruction.

5 Molecule Generation Task

5.1 Experimental Settings

Data Construction To demonstrate the contribution of structural grounding, we constructed a re-

³<https://www.ebi.ac.uk/opsin/>

Tested Model	BLEU	Levenshtein↓	MACCS FTS	RDKit FTS	Morgan FTS	FCD Metric↓	Validity	Exact Match
Inference with domain-specialized language models:								
T5-v1.1-large (fine-tuned)	0.829	10.73	0.840	0.707	0.650	1.895	0.959	0.224
MolT5-large (fine-tuned)	0.837	10.24	0.860	0.738	0.677	1.605	0.975	0.266
ChemDFM-v2-14B (ICL)	0.829	7.371	0.910	0.768	0.716	1.320	0.982	0.414
ChemDFM-v2-14B (CoT)	0.846	6.751	0.913	0.784	0.724	1.372	0.983	0.440
ChemDFM-R-14B (ICL)	0.836	7.442	0.905	0.775	0.712	1.864	0.984	0.434
ChemDFM-R-14B (CoT)	0.857	6.835	0.923	0.811	0.757	1.130	0.979	0.505
Inference with general-purpose language models:								
GPT-4o-1120 (ICL)	0.682	13.84	0.819	0.597	0.529	2.338	0.880	0.142
GPT-4o-1120 (CoT)	0.696	13.26	0.831	0.613	0.553	2.502	0.853	0.133
GPT-4o-1120 (CaS)	0.736	13.16	0.863	0.637	0.603	5.117	0.862	0.182
GPT-4o-1120 (CaS+A)	0.701	13.49	0.856	0.649	0.585	2.664	0.860	0.196
GPT-4o-1120 (CaS+M)	0.721	12.45	0.875	0.689	0.632	4.637	0.839	0.206
GPT-4o-1120 (CaS+A+M)	0.717	12.61	0.868	0.678	0.631	2.464	0.875	0.224
GPT-4o-1120 (ExS Ours)	0.725	12.01	0.871	0.684	0.631	2.525	0.848	0.237
Qwen3-30B-A3B (ICL)	0.485	22.93	0.686	0.435	0.361	4.195	0.843	0.021
Qwen3-30B-A3B (CoT)	0.518	19.38	0.694	0.475	0.434	5.895	0.639	0.067
Qwen3-30B-A3B (CaS)	0.644	15.10	0.840	0.671	0.616	5.098	0.669	0.172
Qwen3-30B-A3B (CaS+A)	0.588	17.70	0.862	0.691	0.650	6.412	0.683	0.218
Qwen3-30B-A3B (CaS+M)	0.583	16.29	0.876	0.731	0.694	8.160	0.603	0.219
Qwen3-30B-A3B (CaS+A+M)	0.623	15.52	0.869	0.708	0.865	4.737	0.670	0.238
Qwen3-30B-A3B (ExS Ours)	0.651	14.72	0.880	0.726	0.683	4.206	0.633	0.257
Claude-Sonnet-4.5 (ICL)	0.858	6.802	0.927	0.778	0.744	3.161	0.930	0.396
Claude-Sonnet-4.5 (CoT)	0.842	6.730	0.929	0.842	0.819	2.107	0.970	0.534
Claude-Sonnet-4.5 (CaS)	0.903	4.728	0.955	0.845	0.808	1.949	0.980	0.530
Claude-Sonnet-4.5 (CaS+A)	0.912	4.201	0.953	0.832	0.816	1.665	0.986	0.545
Claude-Sonnet-4.5 (CaS+M)	0.922	3.418	0.967	0.881	0.851	1.754	0.974	0.612
Claude-Sonnet-4.5 (CaS+A+M)	0.922	3.856	0.964	0.871	0.839	1.705	0.995	0.644
Claude-Sonnet-4.5 (ExS Ours)	0.924	3.139	0.965	0.877	0.862	2.565	0.990	0.683
GPT-5.2-non-thinking (ICL)	0.733	11.83	0.891	0.706	0.656	2.160	0.844	0.202
GPT-5.2-thinking (CoT)	0.779	9.332	0.943	0.825	0.791	1.731	0.819	0.418
GPT-5.2-thinking (CaS)	0.795	7.986	0.958	0.867	0.832	1.823	0.857	0.549
GPT-5.2-thinking (CaS+A)	0.818	7.069	0.957	0.835	0.817	2.911	0.927	0.584
GPT-5.2-thinking (CaS+M)	0.849	6.240	0.962	0.877	0.853	2.106	0.923	0.635
GPT-5.2-thinking (CaS+A+M)	0.872	4.963	0.962	0.880	0.852	0.944	0.963	0.676
GPT-5.2-thinking (ExS Ours)	0.877	4.890	0.964	0.887	0.859	0.966	0.958	0.691

Table 2: Experimental results on the molecule generation task. The lower is better in Levenshtein and FCD Metric. **Exact Match** denotes the identical molecule structure as the ground truth. Blue rows are ablation comparison.

finer subset of ChEBI-20 (Edwards et al., 2022) containing 559 targeted caption-to-molecule samples that demand IUPAC-grounded structural reasoning for substituent attachment at designed locant positions. For each sample, we constructed enriched representations by identifying all referenced scaffolds and substituents, generating IUPAC-aligned SMILES with locant annotations, extracting valence details, and formatting substituents to anchor-aligned SMILES.

Tested Models and Inference Settings We evaluated domain-specialized models: T5-v1.1-large and MolT5-large (Edwards et al., 2022) on their default settings, ChemDFM-v2-14B (Zhao et al., 2024b), ChemDFM-R-14B (Zhao et al., 2025), and general-purpose LLMs: Qwen3-30B-A3B-2507, GPT-4o-1120, GPT-5.2, and Claude-Sonnet-4.5 with different inference configurations (see Appendix G for more implementation details). We compared seven experimental settings to distinguish the contribution of each component. (1) **ICL**: in-context learning with 10-shot examples.

(2) **CoT**: vanilla chain-of-thought for step-by-step reasoning. (3) **CaS**: **C**anonicalized **S**MILES of core molecular scaffolds and substituents are provided. (4) **CaS+A**: CaS with additional **A**nchor-aligned substituents. (5) **CaS+M**: CaS with additional IUPAC-aligned locant-**M**apped scaffold SMILES. (6) **CaS+A+M**: CaS+A with IUPAC-aligned locant-**M**apped scaffold SMILES. (7) **ExS**: Enriched molecule representation: full structure-grounded context including locant-mapped scaffolds, valence information, and anchor-aligned substituents is provided. This comparison design also formulates an ablation study, allowing us to observe models’ capability in utilizing different features under controlled conditions.

Evaluation Metrics We report Exact Match, BLEU, Levenshtein distance (lower is better), FCD measuring distributional similarity (lower is better), Fingerprint Similarities of MACCS, RDKit, and Morgan, and Validity (see metrics details in Appendix F) following common practice (Edwards et al., 2022). Among these, **Exact Match** is the

Model	BLEU	Levenshtein↓	MACCS FTS	RDKit FTS	Morgan FTS	FCD Metric↓	Validity	Exact Match
Inference with the enriched molecule representation:								
Qwen3-30B-A3B	0.651	14.72	0.880	0.726	0.683	4.206	0.633	0.257
GPT-4o-1120	0.725	12.01	0.871	0.684	0.631	2.525	0.848	0.237
Inference with the teacher-guided reasoning protocol:								
Qwen3-30B-A3B	0.701	11.78	0.917	0.781	0.758	2.448	0.726	0.428 [↑67%]
GPT-4o-1120	0.711	11.90	0.893	0.749	0.725	2.712	0.815	0.413 [↑74%]

Table 3: Experimental results on the inductive protocol refinement.

key metric as it directly measures whether the generated molecules are identical to the ground truth.

5.2 Results and Analyses

Table 2 summarizes results on the molecule generation task across all models and conditions. Structure-grounded context consistently improves performance across all general-purpose LLMs: GPT-5.2 achieves the highest exact match accuracy at 0.691, followed by Claude-Sonnet-4.5 at 0.683. Notably, with explicit structure-grounded context, frontier general-purpose LLMs outperform the strong domain-specialized model ChemDFM-R with CoT, while weaker models like Qwen3-30B achieves comparable accuracy to domain-specialized MolT5-large model, suggesting explicit structural grounding can compensate for domain-specific training.

Interestingly, while Claude-Sonnet-4.5 performs the best in the diagnostic tasks across all dimensions (Figure 2), GPT-5.2 achieved the best final results on molecular generation, suggesting that explicit structural context levels the playing field on structural reasoning: when grounding information is provided explicitly, the bottleneck shifts from inherent structural reasoning capacity to general generation capacity.

The ablation results distinguish each component’s contribution and align precisely with our diagnostic findings. For Qwen3-30B, exact match rises from 0.021 (ICL) to 0.067 (CoT) to 0.172 (CaS) to 0.218 (CaS+A) to 0.238 (CaS+A+M) to 0.257 (ExS), showing balanced contributions from all three structural enrichments, consistent with its uniform failure across all diagnostic dimensions. For GPT-5.2, exact match rises from 0.202 (ICL) to 0.418 (CoT) to 0.549 (CaS) to 0.584 (CaS+A) to 0.676 (CaS+A+M) to 0.691 (ExS), with the largest single gain from locant-mapped SMILES, agreeing with locant identification being its primary diagnostic weakness. Weaker models benefit most from enriched representations: Qwen3-30B shows a 284% relative gain from CoT to ExS, suggesting explicit

structural grounding is particularly valuable when models lack inherent structural reasoning capabilities. Fine-grained error analysis further confirms that grounding components systematically reduce their corresponding failure modes (Appendix Figure 6).

Although previous work emphasizes validity (Tao et al., 2025), exact match is the most critical in chemistry: valid yet nonidentical structure still means different molecules; exact molecular identity is required. The slightly fluctuated validity reflects models attempting more complex structures rather than conservative generation; **exact match improvements present the actual gains**. These results validate our core thesis: providing structure-grounded context that explicitly encodes locant positions, anchoring points, and valency information substantially improves molecular reasoning across model scales.

5.3 Improving Weaker Models via Inductive Protocol Synthesis

Despite consistent gains from our approach, weaker models still largely underperform. Analysis of reasoning traces reveals that they can correctly identify scaffolds and parse modification instructions, but fail at syntactic execution such as endless thinking without a resolution (Figure 10) or losing track of structural state during generation process (Appendix E). This suggests the failure lies in procedural organization, promoting procedural guidance as a targeted solution.

We thus leveraged an instruction induction strategy (Madaan et al., 2023; Xiao et al., 2025) using the teacher-guided protocol. The reasoning traces from weaker models on five failed examples in a hold-out set were provided to Claude-Sonnet-4.5, which was then tasked with identifying the failure patterns and generating corrective guidance. This process yields a decomposed protocol covering locant-map understanding, modification parsing, structure updating and constraint validation (Appendix Figure 13). Since the hold-out set shares

Tested Model (Representation)	BLEU	Levenshtein↓	MACCS FTS	RDKit FTS	Morgan FTS	FCD Metric↓	Validity	Exact Match
GPT-4o-1120 (Atom-mapped)	0.710	13.13	0.856	0.654	0.595	5.369	0.821	0.139
Qwen3-30B-A3B (Atom-mapped)	0.557	18.38	0.838	0.667	0.630	8.169	0.538	0.154
GPT-5.2-thinking (Atom-mapped)	0.799	8.020	0.952	0.847	0.808	2.435	0.886	0.507
Claude-Sonnet-4.5 (Atom-mapped)	0.886	5.532	0.952	0.844	0.810	1.745	0.955	0.527
ChemDFM-X-13B (2D Graph)	0.854	8.453	0.909	0.770	0.731	1.377	0.982	0.388
GPT-4o-1120 (SELFIES)	0.393	28.94	0.540	0.344	0.289	10.17	0.889	0.045
GPT-5.2-thinking (SELFIES)	0.237	44.36	0.560	0.392	0.343	10.77	0.746	0.109
Claude-Sonnet-4.5 (SELFIES)	0.502	21.92	0.674	0.454	0.402	8.175	0.958	0.144
GPT-4o-1120 (MolJSON)	0.727	12.32	0.871	0.671	0.622	3.758	0.857	0.197
GPT-5.2-thinking (MolJSON)	0.861	5.597	0.961	0.859	0.828	1.813	0.945	0.572
Claude-Sonnet-4.5 (MolJSON)	0.896	4.756	0.948	0.831	0.795	1.810	0.985	0.502

Table 4: Comparison with other molecule representations on the molecule generation task.

the property of our constructed test sets, the resulting protocol provides targeted yet general guidance rather than instance-specific instructions. This leads to substantial improvement (Table 3): Qwen3-30B improves from 25.7% to 42.8% exact match and GPT-4o rises from 23.7% to 41.3%, nearly doubling exact match accuracy for both models without any parameter updates. Fine-grained error analysis shows that the protocol specifically targets the residual locant positioning failures during execution (Appendix H, Figure 6). These results demonstrate that explicit procedural scaffolding addresses the structural reasoning gap, and that inductive instruction refinement is a flexible path for bridging performance gaps across model scales.

5.4 Additional Comparison with Other Molecular Representations

We conducted additional comparisons with some other molecular representations and the results are summarized in Table 4. (1) **SMILES with arbitrary atom numbers**. The basic syntax of “atom-mapped SMILES” is briefed in Appendix A, and RDKit is used to generate the atom-mapped annotations. With arbitrary atom numbers, the generation accuracy degrades to values below CaS in Table 2 (GPT-4o, 0.182, Qwen3-30B-A3B, 0.172; Claude-Sonnet-4.5, 0.53; GPT-5.2, 0.549). This suggests the arbitrary atom numbers confuses models, strongly supporting our findings that structure-grounded information is essential for models to perform structure-aware molecule manipulation. (2) **Multimodal encoding**. We use the multimodal model ChemDFM-X (Zhao et al., 2024a) and enrich the text input with the 2D molecular graphs. Adding the graph modality does not bring any performance gain compared with its text-only counterpart, suggesting the structural information the graph carries is not recovered for molecule generation. (3) **SELFIES**. We found that all tested

models performed poorly with SELFIES, possibly due to their limited exposure to SELFIES during pre-training. Qwen3-30B-A3B frequently failed to produce valid outputs, so we excluded it from the reported results. (4) **Adjacency-list JSON variant**. We also tested the MolJSON (Runcie et al., 2026) on GPT-4o, GPT-5.2, and Claude-Sonnet-4.5. The MolJSON scheme offers some structure-grounded information such as **bonds and neighboring atoms**. It results in better accuracy than CaS with GPT models, but not with Claude-Sonnet-4.5, indicating additional guidelines are necessary to better utilize such representations across models. The overall accuracy is lower than CaS + M inference setting, suggesting locant-mapped structural information is key for accurate SMILES manipulation. Together, this experiment provides further evidence on the significance of structural grounding for molecular understanding and manipulation.

6 Conclusion

We systematically diagnosed the grounding problem in molecular reasoning. Decomposing it along three IUPAC-grounded dimensions, our diagnostic reveals universal but non-uniform failures: locant identification is the most challenging, revealing an inherent limitation of linearized representations, whereas valency failures trace back to structural parsing gaps. By directly enriching each failure mode with its corresponding structural context, general-purpose LLMs achieved nearly 70% accuracy on molecular generation. A teacher-guided reasoning protocol effectively boosted weaker models’ performance without parameter updates. These results establish explicit structural grounding as a principled and training-free path toward reliable molecular reasoning, and suggest that diagnosing failures along theory-grounded dimensions is a productive direction for scientific NLP more broadly.

Limitation

In this work, our experiments were conducted on a refined subset of the ChEBI-20 with the focus on molecule manipulation, which is a challenging and representative task for structure-aware molecular understanding and generation. Its generalizability to more complex chemical generative tasks like retro-synthesis remains to be investigated in future studies. Additionally, incorporating the enriched representation relies on retrieving and validating structural context; performance may degrade for novel molecules where specific scaffold information or IUPAC numbering is unavailable in the knowledge base. Finally, the current state-of-the-art performance relies on access to proprietary frontier LLMs, which limits its applicability in environments where such models are not available.

Ethics Statement

This work aims to advance language model capabilities for scientific and educational applications. While we acknowledge the dual-use risks inherent in generative chemistry, our research is strictly intended for benign purposes and serves as a proof-of-concept for the proposed methodology. Our experiments rely exclusively on publicly available molecular datasets and do not optimize for toxicity or harmful properties.

Acknowledgements

This research is supported by the Nanyang Technological University, Singapore, and by the Agency for Science, Technology and Research (A*STAR), Singapore. We thank the anonymous reviewers for their precious feedback to help improve and extend this piece of work.

References

- Anthropic. 2025. [Claude sonnet 4.5 system card](#).
- Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.
- Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, and 1 others. 2025. Reasoning models don’t always say what they think. *arXiv preprint arXiv:2505.05410*.
- Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. 2023. Unifying molecular and textual representations via multi-task language modelling. In *International Conference on Machine Learning*, pages 6140–6157. PMLR.
- 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 preprint arXiv:2407.21783*.
- Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2021. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*.
- Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. 2022. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413.
- Carl Edwards, ChengXiang Zhai, and Heng Ji. 2021. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607.
- Veronika Ganeeva, Kuzma Khrabrov, Artur Kadurin, and Elena Tutubalina. 2025. Measuring chemical llm robustness to molecular representations: a smiles variation-based framework. *Journal of Cheminformatics*, 17(1):1–15.
- Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, and 1 others. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. 2024. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169.
- Yunhui Jang, Jaehyung Kim, and Sungsoo Ahn. 2025a. Improving chemical understanding of llms via smiles parsing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15694–15709.
- Yunhui Jang, Jaehyung Kim, and Sungsoo Ahn. 2025b. Structural reasoning improves molecular understanding of llm. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21016–21036.

- Lei Jiang, Shuzhou Sun, Biqing Qi, Yuchen Fu, Xiaohua Xu, Yuqiang Li, Dongzhan Zhou, and Tianfan Fu. 2025. Chem3dLLM: 3d multimodal large language models for chemistry. *arXiv preprint arXiv:2508.10696*.
- 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.
- Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. 2020. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024.
- Miguelangel Leon, Yuriy Perezhohin, Fernando Peres, Aleš Popovič, and Mauro Castelli. 2024. Comparing smiles and selfies tokenization for enhanced chemical language modeling. *Scientific Reports*, 14(1):25016.
- Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. 2024. Towards 3d molecule-text interpretation in language models. In *International Conference on Learning Representations*, volume 2024, pages 17352–17371.
- Chang Liao, Yemin Yu, Yu Mei, and Ying Wei. 2024. From words to molecules: A survey of large language models in chemistry. *arXiv preprint arXiv:2402.01439*.
- Pengfei Liu, Jun Tao, and Zhixiang Ren. 2025. A quantitative analysis of knowledge-learning preferences in large language models in molecular science. *Nature Machine Intelligence*, 7(2):315–327.
- Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2023. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15623–15638.
- 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.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.
- Siddharth M Narayanan, James D Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G Rodrigues, and Andrew D White. 2025. Training a scientific reasoning model for chemistry. *arXiv preprint arXiv:2506.17238*.
- OpenAI. 2025. Gpt-5.2 system card. *Technical Report*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer. 2018. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741.
- David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754.
- Nicholas T Runcie, Fergus Imrie, and Charlotte M Deane. 2026. Molecular representations for large language models. *arXiv preprint arXiv:2605.01822*.
- Wen Tao, Jing Tang, Alvin Chan, Bryan Hooi, Baolong Bi, Nanyun Peng, Yuansheng Liu, and Yiwei Wang. 2025. How to make large language models generate 100% valid molecules? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26576–26591.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.
- Weida Wang, Benteng Chen, Di Zhang, Wanhao Liu, Shuchen Pu, Ben Gao, Jin Zeng, Xiaoyong Wei, Tian-shu Yu, Shuzhou Sun, and 1 others. 2025. Chemr: Learning to reason as a chemist. *arXiv preprint arXiv:2510.16880*.
- 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.
- David Weininger. 1988. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36.
- Emily Xiao, Yixiao Zeng, Ada Chen, Chin-Jou Li, Amanda Bertsch, and Graham Neubig. 2025. Prompt-mii: Meta-learning instruction induction for llms. *arXiv preprint arXiv:2510.16932*.

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 arXiv:2505.09388*.

Qiang Zhang, Keyan Ding, Tianwen Lv, Xinda Wang, Qingyu Yin, Yiwen Zhang, Jing Yu, Yuhao Wang, Xiaotong Li, Zhuoyi Xiang, and 1 others. 2025. Scientific large language models: A survey on biological & chemical domains. *ACM Computing Surveys*, 57(6):1–38.

Zihan Zhao, Bo Chen, Jingpiao Li, Lu Chen, Liyang Wen, Pengyu Wang, Zichen Zhu, Danyang Zhang, Yansi Li, Zhongyang Dai, Xin Chen, and Kai Yu. 2024a. *Chemdfm-x: towards large multimodal model for chemistry*. *Science China Information Sciences*, 67(12):220109.

Zihan Zhao, Bo Chen, Ziping Wan, Lu Chen, Xuanze Lin, Shiyang Yu, Situo Zhang, Da Ma, Zichen Zhu, Danyang Zhang, and 1 others. 2025. *Chemdfm-r: A chemical reasoning llm enhanced with atomized chemical knowledge*. *arXiv preprint arXiv:2507.21990*.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, and 1 others. 2024b. *Chemdfm: A large language foundation model for chemistry*. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*.

A Atom-Mapped SMILES Syntax

Standard SMILES (Weininger, 1988) strings represent molecules as sequences of characters, where atoms are nodes and bonds are edges. However, standard SMILES do not enforce a unique mapping between the string characters and the atoms in the molecular graph, as the traversal order can vary. Atom-mapped SMILES extend this notation by assigning a unique integer identifier to each atom, ensuring a consistent reference for every position in the structure.

In atom-mapped SMILES, an atom is represented within brackets ‘[]’, containing the atomic symbol followed by a colon ‘:’ and the integer map number. For instance, the standard SMILES of Ethanol is CCO, and its Atom-mapped SMILES is [CH3:1][CH2:2][OH:3]. This notation explicitly maps the methyl carbon to index 1, the methylene carbon to index 2, and the hydroxyl oxygen to index 3. By assigning unique identifiers, it enables models to distinguish between chemically identical atoms, track specific positions across reaction steps, and link textual descriptions like “C-2” directly to their corresponding sequence tokens.

Given that SMILES representations are non-unique and often appear in non-canonical forms within pretraining data, we conducted input randomization to testing the model robustness. Specifically, **we randomly reordered the atom-mapped SMILES in our diagnostic tasks**, forcing the models to interpret the explicit atom mapping rather than relying on fixed canonical sequences.

B IUPAC Locant Mapping Construction

We use OPSIN to interpret the IUPAC name and assign all explicit locants (in the CML format), naturally handling complex cases like fused or bridged systems (e.g., 4a, 8a). We then use RDKit to initialize the molecule object and incorporate the extracted OPSIN locants into it. Because RDKit’s atom mapping only accepts integers, we generate an atom-indexed SMILES using temporary integer placeholders. Finally, we apply a regular expression substitution to replace these temporary IDs with the exact locants in the final string (e.g., mapping a temporary [c:1] to its true [c:4a]). While this procedure is mechanistically valid by construction, a domain expert reviewed 50 randomly selected samples, validated them using ChemDraw, and confirmed that all of them were correct (100% conversion accuracy).

C Molecule Named Entity Extraction

In our experiments, we use LLMs (e.g., GPT-4o) to extract molecular named entities from captions (e.g., base scaffold and substituents), which is more flexible than template-based entity matching. Here is one instruction example: “*Extract from the following molecule caption the base molecular scaffold(s) and any named substituents. Output as JSON with keys “scaffold” and “substituents” (an array of objects with exact names).*” To ensure validity, we verify extracted entity names by checking their occurrence in the original caption and filtering out those absent from common chemical databases (e.g., PubChem). We quantified the entity extraction accuracy and retrieval coverage. For entity-extraction accuracy, a domain expert manually audited GPT-4o’s extractions on 100 captions and confirmed 96% accuracy. As this step is text extraction over explicitly named scaffolds and substituents, high accuracy is expected. For retrieval coverage: around 95% of extracted entities resolve to SMILES, with almost 100% scaffold retrieval. The failure concentrates on substituent retrieval and

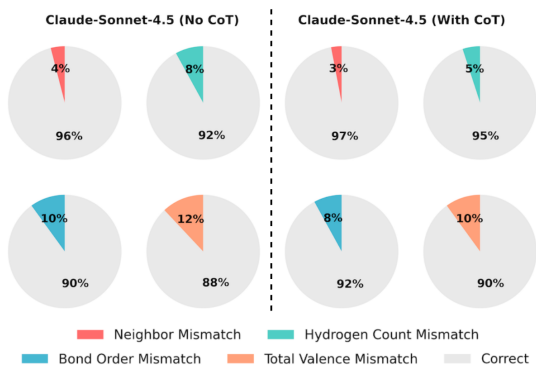


Figure 5: Error type analysis on the valence and neighbor labeling evaluation (Claude-Sonnet-4.5).

Models	Mean	Std. Dev↓
GPT-4o-1120	0.21	0.012
Qwen3-30B-A3B	0.22	0.016

Table 5: Mean exact-match accuracy and standard deviation over 5 independent runs of locant identification.

anchoring point identification. As the SMILES retrieval is done by standard cheminformatics tools, we treat this as a reliable preprocessing component.

D Additional Experimental Results of Diagnostic Tasks

(1) The error-type analysis on valence and neighbor labeling for Claude-Sonnet-4.5 is shown in Figure 5. This is a supplementary data for Figure 3.

(2) For the diagnostic task, we didn’t expect sensitivity towards the few-shot examples since the examples were randomly sampled from a held-out set with no scaffold overlap with the test instances. To verify this directly, we re-ran locant identification 5 times per model; each run independently sampled 100 test instances, 3 few-shot demonstrations, and a random atom-index relabeling of the input SMILES. We report mean exact-match accuracy and its standard deviation across the multiple runs in Table 5. The low variance confirms the identification is robust to the few-shot demonstrations and atom-index relabeling.

E Reasoning Traces as Interpretable Interfaces

Although recent research suggests that chain-of-thought explanations may lack faithfulness, often providing post-hoc justifications that obscure underlying decision-making mechanisms (Turpin et al., 2023; Chen et al., 2025), we analyze reasoning traces to understand how models prioritize

and contextualize structural information. While generated rationales may not perfectly reflect internal causal operations, they explicitly expose the model’s procedural planning and information distillation. This is crucial in scientific domains like molecular reasoning, where the validity of the derivation steps is as important as the final answer. Furthermore, natural language traces serve as a vital interface for human-AI collaboration, offering the interpretability necessary for domain experts to identify structural misconceptions and verify the chemical logic driving model decisions.

Our Observation As the examples shown in Figure 10, Figure 11, and Figure 12, weaker models (e.g., Qwen3-30B and GPT-4o) are able to initiate the molecule generation task with reasonable logic planning: basic steps, and analyzing the input and requirements. However, unlike stronger models like GPT-5.2 and Claude-Sonnet-4.5, they exhibit significant fragility during intermediate reasoning steps. In particular, we observed patterns of recursive failure, where the Qwen3-30B failed into “endless thinking” loops, it repeatedly re-evaluating the same structural constraint without progressing to a resolution, or lose track of ring numbers as the SMILES string evolves. This divergence highlights that even we provided requisite chemical knowledge, weaker models may lack the robust state-tracking capabilities needed to sustain precise structural modifications over long reasoning horizons.

F Molecule Generation Metrics

As common practice (Edwards et al., 2022), we use following metrics to evaluate model-generated molecules. We canonicalize the model-generated and ground truth SMILES via RDKit (where the molecule string is parseable) to reflect structure identity beyond string similarity.

BLEU (Papineni et al., 2002) measures the textual similarity between the generated and reference SMILES strings by calculating overlap of n -grams. **Levenshtein Distance** quantifies the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform the generated SMILES string into the reference string. **Fingerprint Tanimoto Similarities** measure structural similarity by calculating the Jaccard index between the boolean fingerprint vectors (MACCS, RDKit, or Morgan) of the generated and reference molecules (Rogers and Hahn, 2010).

Fréchet ChemNet Distance (FCD) (Preuer et al., 2018) evaluates the distance between the distribution of generated molecules and real molecules in a high-dimensional feature space.

Validity reports the percentage of generated SMILES sequences that follow correct chemical syntax and valency rules, resulting in a parseable molecular structure.

Exact Match verifies whether the generated molecule is chemically identical to the ground truth after canonicalizing both structures to handle SMILES variations.

G Model Selection and Inference Configurations

For the three diagnostic evaluation tasks, we selected general-purpose models: Meta-Llama-3.1-8B (Dubey et al., 2024), Gemma-3-12B (Team et al., 2025), Qwen3-30B-A3B-2507 (Yang et al., 2025), GPT-4o-1120 (Hurst et al., 2024), GPT-5.2 (OpenAI, 2025), Claude-Sonnet-4.5 (Anthropic, 2025), and a state-of-the-art domain-specialized model ChemDFM-v2.0-14B (Zhao et al., 2025). We include both open-source models for better reproducibility, and proprietary models to establish performance upper bounds on general-purpose models. ChemDFM-v2.0-14B serves as a strong baseline as domain-specialized LLMs, allowing us to evaluate the effectiveness of in-domain training.

For the molecule generation task, we selected strong domain-specialized models including T5-v1.1-large and MolT5-large (Edwards et al., 2022), ChemDFM-v2-14B (Zhao et al., 2024b), and ChemDFM-R-14B (Zhao et al., 2025), along with general-purpose LLMs including Qwen3-30B-A3B, GPT-4o-1120, Claude-Sonnet-4.5, and GPT-5.2 (w/ default thinking effort). We particularly focus on general-purpose LLMs to test whether explicit structural grounding can compensate for the lack of massive domain-specific training. For non-thinking mode, we employed temperature near 0 for deterministic generation, and default temperature is used for thinking modes. We applied the “*Let’s think step by step*” as vanilla chain-of-thought prompt. For proprietary LLMs, we use their official API service without web-search function calling and other features.

Moreover, we exclude Meta-Llama-3.1-8B and Gemma-3-12B from the molecule generation evaluation because preliminary experiments revealed that their performance remained very low even

when provided with enriched molecular context, suggesting that a minimum capability threshold is required to effectively leverage structural grounding. For the ChemDFM models, we observed that their domain-specific instruction tuning led to rigid response patterns, resulting in poorer instruction-following compared to general-purpose models when presented with various output formats.

H Fine-Grained Error Analysis

To further evaluate the effectiveness of our proposed method and gain deeper insights into failure modes in the molecule generation task, we implemented a fine-grained via LLM-as-a-judge (using GPT-5.2 with thinking mode). We focus on locant positioning and anchoring recognition errors, as valency and bonding errors typically result in invalid SMILES and are captured by the validity metric; isolating them as a separate category would conflate structural reasoning failures with syntactic execution errors. More specifically, we count the failure modes of two types: (1) **Locant Positioning Errors**: the model targets an incorrect numerical position or relative geometry on the base molecular scaffold, and (2) **Anchoring Recognition Errors**: the model incorrectly identifies which atom on the substituent itself should form the connection. This allows us to measure how enriched structural context mitigates symbol-to-graph translation errors.

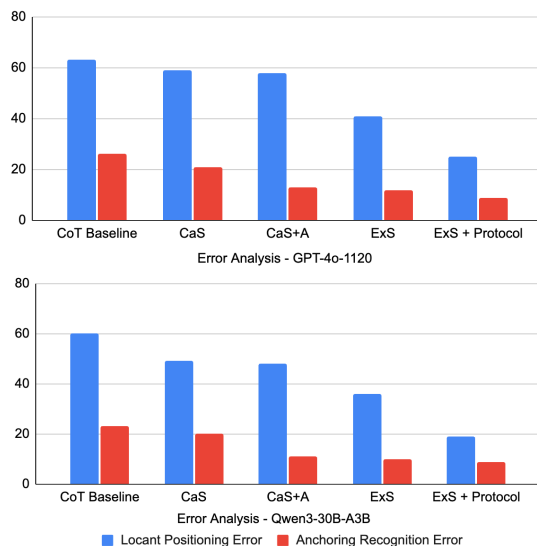


Figure 6: Fine-grained error analysis results. The Y-axis denotes the percentage of errors over tested samples.

Figure 6 presents the fine-grained error analysis for GPT-4o-1120 and Qwen3-30B-A3B under five inference settings, since they provide

full chain-of-thought reasoning traces. First, anchoring errors respond readily to structural enrichment: adding anchoring information (CaS+A) nearly halves them. This indicates that anchoring is an information-retrieval problem solvable by explicit context, whereas locant reasoning additionally requires sustained positional tracking during generation. Second, we observed that the locant positioning failures under ExS are not knowledge gaps but execution failures. As shown in Figure 13, compare with stronger models (e.g., GPT-5.2, Claude-Sonnet-4.5), models with less reasoning efforts (e.g., GPT-4o, Qwen3-30B) understand the structural context but lose track of them over long generation horizons. Moreover, both models converge to comparable error rates under ExS + Teacher-guided Protocol, suggesting that procedural guidance effectively scaffolds structural reasoning beyond what enriched representations alone provide.

I Task Instructions and Model Generation Examples

We list some examples of task-specific instruction for the three diagnostic evaluation tasks: locant identification (Figure 7), anchoring point recognition (Figure 8), valence and neighbor labeling (Figure 9).

We also show instruction and model outputs of the molecule generation task: GPT-4o output of chain-of-thought reasoning given canonicalized base scaffold (Figure 11), Claude-Sonnet-4.5 output of chain-of-thought reasoning given enriched molecule representation (Figure 12).

Moreover, we demonstrate one example of the failure pattern recognition of GPT-4o reasoning traces by a stronger teacher model (e.g., Claude-Sonnet-4.5), as shown in Figure 13. The teacher-guided protocol is used as input to refined the task-instruction for weaker models (e.g., GPT-4o and Qwen3-30B-A3B), helping them better leverage the enriched molecule representations.

J Licenses and Terms of Use

Datasets and models used in this work are publicly available under open licenses. ChEBI-20 is distributed under the Creative Commons Attribution license (<https://www.ebi.ac.uk/licencing/>). PubChem data is publicly available through the NCBI and is freely accessible for research use. RDKit is distributed under the BSD license

(<https://github.com/rdkit/rdkit/blob/master/license.txt>). For proprietary models (e.g., GPT-4o, GPT-5.2, Claude-Sonnet-4.5), we accessed them via official APIs in compliance with their respective terms of service, and no model weights were redistributed. Open-source models (e.g., Meta-Llama-3.1-8B, Gemma-3-12B, Qwen3-30B-A3B, Mol-T5, ChemDFM-v2.0-14B, ChemDFM-R-14B, ChemDFM-X-13B) are used in accordance with their respective licenses. The local deployed models are running on the Nvidia A100 GPUs with 80G memory.

K LLM Usage Declaration

We used ChatGPT and Grammarly solely for grammar checking and correction when writing the manuscript. All scientific content, experimental designs, results, and conclusions were written and verified by the authors.

You are an expert chemist specializing in molecular representation and structure. Given a molecule SMILES, identify the atom that should be assigned Locant 1 according to IUPAC nomenclature rules. Please label the Locant 1 atom as [Element:1]. Do not map or change other atoms.

Think step by step, then give me the output as the following json format:

```
{
  "Output": "labeled_smiles"
}
```

Here are some examples:

===

IUPAC name: tetrahydroisoquinolinium

Input SMILES: c1ccc2c(c1)C[NH2+]CC2

Input Template: [c:H:X]1[c:H:X][c:H:X][c:X]2[c:X]([c:H:X]1)[CH2:X][NH2+:X][CH2:X][CH2:X]2

```
{
  "Output": "[c:H:X]1[c:H:X][c:H:X][c:X]2[c:X]([c:H:X]1)[CH2:1][NH2+:X][CH2:X][CH2:X]2"
}
```

... ..

===

Here is the sample to label:

IUPAC name: 1-naphthoate

Input SMILES: c12cccc(C(=O)[O-])c1cccc2

Input Template: [c:X]12[c:H:X][c:H:X][c:H:X][c:X](C(=O)[O-])[c:X]1[c:H:X][c:H:X][c:H:X][c:H:X]2

===

Model prediction:

[c:1]12[c:H:99][c:H:99][c:99](C(=O)[O-])[c:99]1[c:H:99][c:H:99][c:H:99][c:H:99]2

Ground truth:

[c:4]12[c:H:4][c:H:3][c:H:2][c:1](C(=O)[O-])[c:8]1[c:H:8][c:H:7][c:H:6][c:H:5]2

Correctness: False

Reason: Wrong Position (Mapped to GT atom 0, Expected 4)

Figure 7: Diagnostic evaluation examples: the locant identification task.

You are an expert chemist specializing in molecular representation and substructures. Identify the anchoring point of a specific substituent group within its parent molecule structure.

Input Molecule: c1ccccc1[CH]C

Atom-Mapped SMILES: [c:H:1]1[c:H:5][c:H:4][c:H:7][c:H:2][c:8]1[CH:6][CH3:3]

Question: If the molecule above acts as the substituent "(1R)-1-phenylethyl", which atom (identified by its Map ID number) serves as the attachment point to the rest of the scaffold?

Think step by step, then give me the integer ID of the anchoring atom in the following json format:

```
{
  "atom_id": "xxx"
}
```

===

Model reasoning trace (CoT):

To determine the anchoring point of the substituent group "(1R)-1-phenylethyl" within the given molecule, we need to analyze the Atom-Mapped SMILES representation step by step.

Step 1: Understanding the substituent

The substituent "(1R)-1-phenylethyl" consists of a phenyl ring (benzene) attached to a chiral ethyl group. Specifically, the attachment point to the scaffold will be the carbon atom in the ethyl group that connects directly to the phenyl ring.

Step 2: Analyzing the Atom-Mapped SMILES

The Atom-Mapped SMILES is:

```

[c:H:1]1[c:H:5][c:H:4][c:H:7][c:H:2][c:8]1[CH:6][CH3:3]

```

- `[c:H:1]1` starts the phenyl ring with atom ID 1.

- `[CH:6]` represents the ethyl group's first carbon atom, which is directly bonded to the phenyl ring.

- `[CH3:3]` represents the methyl group attached to `[CH:6]`.

Step 3: Identifying the anchoring atom

The anchoring atom is the carbon atom in the ethyl group that connects the substituent to the rest of the scaffold. This atom is `[CH:6]`, as it is the first carbon in the ethyl group bonded to the phenyl ring.

Final Answer:

The anchoring atom ID is `6`.

Output in JSON format:

```json

```
{
 "atom_id": "6"
}
```

```

===

Model Prediction: 6

Ground Truth: 6

Correctness: True

Figure 8: Diagnostic evaluation examples: the anchoring point recognition task.

You are an expert chemist specializing in molecular representation and substructures.
 Analyze the valence of the Carbon atom with Map ID 4 (i.e., the `[:4]` shown in the given SMILES).

Input Molecule: c1c2ccccc2c(C(=O)[O-])cc1
 Atom-mapped SMILES: [CH:7]1[c:5]2[CH:6][CH:1][CH:2][CH:3][:4]2[c:10]([C:11](=[O:12])[O-:13])[CH:9][CH:8]1
 The `[:4]` is a Carbon atom.

Based on the structure and connectivity (e.g., rings, bonds, branches) shown in the SMILES, break down the valence into its components by filling in the following fields:

Neighbors & Bond Orders: List its immediate neighbors (with atom ID) and the bond order (1=Single, 2=Double, 3=Triple, 1.5=Aromatic) connecting to it.
 Hydrogen Count: Imply the number of hydrogen atoms to satisfy the valence.
 Total Valence: Show the valence sum.

Directly give me the output as the following json format:

```
{
  "target_atom_id": "4",
  "atom_type": "Carbon",
  "neighbor_bonds": [{"neighbor_map_id": "xxx", "atom_type": "xxx", "bond_type": "xxx", "bond_order": xxx}, ...], # List immediate neighbors
  "hydrogen_count": xxx,
  "total_valence": xxx,
}
```

====

Model reasoning trace:
 Let's analyze the given molecule step by step.
 ### Input Molecule:
 Atom-mapped SMILES: [CH:7]1[c:5]2[CH:6][CH:1][CH:2][CH:3][:4]2[c:10]([C:11](=[O:12])[O-:13])[CH:9][CH:8]1
 The target atom is `[:4]`, which is a **Carbon atom**. Let's determine its connectivity and valence.

Step 1: Analyze the neighbors and bond orders of `[:4]`.
 From the atom-mapped SMILES:
 - `[c:5]` is connected to `[:4]` via an **aromatic bond** (bond order = 1.5).
 - `[CH:10]` is connected to `[:4]` via an **aromatic bond** (bond order = 1.5).
 Thus, the neighbors of `[:4]` are:
 1. `[c:5]` (Carbon) with an aromatic bond (bond order = 1.5).
 2. `[CH:10]` (Carbon) with an aromatic bond (bond order = 1.5).

Step 2: Hydrogen count for `[:4]`.
 The Carbon atom `[:4]` is part of an aromatic ring system. Each aromatic Carbon typically has one hydrogen atom unless otherwise specified. In this case, there is no explicit hydrogen on `[:4]`, meaning it has **0 hydrogens**.

Step 3: Total valence calculation.
 The total valence is calculated as the sum of the bond orders to neighbors and the number of hydrogens:
 - Bond order to `[c:5]`: 1.5
 - Bond order to `[CH:10]`: 1.5
 - Hydrogen count: 0
 Total Valence = 1.5 + 1.5 + 0 = 3

Model prediction:

```
{'target_atom_id': '4', 'atom_type': 'Carbon',
 'neighbor_bonds': [{'neighbor_map_id': '5', 'atom_type': 'Carbon', 'bond_type': 'aromatic', 'bond_order': 1.5},
 {'neighbor_map_id': '10', 'atom_type': 'Carbon', 'bond_type': 'aromatic', 'bond_order': 1.5}],
 'hydrogen_count': 0, 'total_valence': 3}
```

Gold reference:

```
{'target_atom_id': 4, 'atom_type': 'C', 'hydrogen_count': 0,
 'neighbor_bonds': [{'neighbor_map_id': 3, 'atom_type': 'C', 'bond_type': 'AROMATIC', 'bond_order': 1.5},
 {'neighbor_map_id': 10, 'atom_type': 'C', 'bond_type': 'AROMATIC', 'bond_order': 1.5},
 {'neighbor_map_id': 5, 'atom_type': 'C', 'bond_type': 'AROMATIC', 'bond_order': 1.5}], 'total_valence': 4.5, 'is_boring': False}
```

Evaluate result: Atom 4: FAIL
 Prediction Errors: `['total_valence_mismatch', 'neighbor_mismatch']`

Figure 9: Diagnostic evaluation examples: the valence and neighbor labeling task.

Wait – the `1` in `c1` is the start, and `2` is the end of the first ring, but then `cccc12` starts a new ring.

Actually, the correct interpretation is:

- `c1` → start of ring A (positions 1–8)
- `C(=O)[0-]` → attached to the first carbon (position 1)
- Then `c` → position 2
- `c` → position 3
- `c` → position 4
- `c` → position 8
- `2` → closes ring A (from 8 to 1) → so ring A is complete
- Then `cccc12` → starts ring B: `c` (position 8), `c` (position 7), `c` (position 6), `c` (position 5), `c` (position 4), then `1` → connects to position 1, and `2` → connects to position 8? But `2` is already used.

This is **invalid** because ring closure labels are reused.

Wait – actually, `c1(C(=O)[0-])cccc2cccc12` means:

- `c1` → atom 1
- `(C(=O)[0-])` → attached to atom 1
- Then `c` → atom 2
- `c` → atom 3
- `c` → atom 4
- `c` → atom 8
- `2` → closes ring from atom 8 back to atom 1 → so ring A is closed via `2`?
- Then `cccc12` → this is a second ring: `c` (atom 8), `c` (atom 7), `c` (atom 6), `c` (atom 5), `c` (atom 4), then `1` → connects to atom 1, and `2` → connects to atom 8? But atom 8 is already connected.

This is **incorrect**.

Actually, the correct SMILES for 1-naphthoate (deprotonated) is:

c1ccc2c(c1)C(=O)[O-] – but that's for 2-naphthoate.

Wait – let's check a reliable source.

Standard SMILES for **1-naphthoic acid** is:

c1ccc2c(c1)C(=O)O

For the **deprotonated form (1-naphthoate)**: c1ccc2c(c1)C(=O)[O-]

But that corresponds to:

- Ring A: c1ccc2c(c1) → positions 1,2,3,4,5,6,7,8? Let's map:
- `c1` → position 1
- `c` → 2
- `c` → 3
- `c` → 4
- `2` → ring closure to form the second ring
- `c(c1)` → carbon 5 attached to ring A (position 1) and to the next carbon
- `c` → 6
- `c` → 7
- `c` → 8
- `c1` → connects back to position 1? But `c1` is already used.

Actually, the correct SMILES for **1-naphthoate** is:

c1ccc2c(c1)C(=O)[O-]

This means:

- `c1` → start of ring A (position 1)
- `c` → 2
- `c` → 3
- `c` → 4
- `2` → closes ring A to form the fused ring (positions 1–4, then 2–5, etc.)
- `c(c1)` → carbon 5 attached to position 1 and to the next carbon
- `c` → 6
- `c` → 7
- `c` → 8
- `C(=O)[O-]` → attached to position 5? No – `c(c1)` means a carbon attached to position 1 and to the next atom.

Wait – c1ccc2c(c1) means:

- `c1` → atom 1
- `2` → closes ring from 4 to 1? But `2` is the ring label.

truncated for better readability

No – the `C(=O)[O-]` is attached to the `c` that is part of the `c(c1)` group.

Actually, the correct SMILES for **1-naphthoic acid** is:

c1ccc2c(c1)C(=O)O

This means:

- Ring A: c1ccc2c(c1) → positions 1,2,3,4, and ring closure `2` from 4 to 1
- Then `c(c1)` → carbon 5 attached to position 1
- Then `c` → 6
- `c` → 7
- `c` → 8
- `C(=O)O` → attached to carbon 5? No – the `C(=O)O` is attached to the `c` in `c(c1)` – so carbon 5 is the one with the carboxyl.

truncated for better readability

The correct SMILES for **1-naphthoic acid** is:

c1ccc2c(c1)C(=O)O – but that's not correct.

Wait – let's use a standard source....

truncated for better readability

Figure 10: One example of the endless thinking of Qwen3-30B-A3B-2507.

You are an expert in computational chemistry and molecule structure.
Your task is to translate a textual molecule description into a valid SMILES string.
The caption describes the molecule's core scaffold, substituents, stereochemistry, and sometimes its derivation and properties.

Here are some SMILES notation examples:
===
Molecule Caption: The molecule is a member of the class of pterocarpanes carrying two hydroxy substituents at positions 6a and 9 as well as prenyl and methoxy substituents at positions 2 and 3 respectively. It has a role as a plant metabolite. It is a member of pterocarpanes, a member of phenols and an aromatic ether.
SMILES Notation: COC1cc2c(cc1CC=C(C)C)[C@@H]10c3cc(OC)ccc3[C@]1(O)COC2
...
===

Here is some useful information:
===
1,4-benzoquinone: C1(=O)C=CC(=O)C=C1
hydroxy: [OH-]
methoxy: C[O]
methyl: [CH3]
2,3-dihydroxy-5-methyl-1,4-benzoquinone: OC=1C(C=C(C(C1O)=O)C)=O
===

Task: provide the SMILES notation from the following molecule caption.
Molecule Caption: The molecule is a member of the class of monohydroxy-1,4-benzoquinones that is 1,4-benzoquinone which is substituted by a methoxy, hydroxy and methyl group at positions 2,3 and 5, respectively. It is a mycotoxin isolated from *Aspergillus fumigatus*. It has a role as an *Aspergillus* metabolite and a mycotoxin. It derives from a 2,3-dihydroxy-5-methyl-1,4-benzoquinone.

Use the above information and think step by step.
Then, validate your prediction by checking the SMILES validity including rings, branches, aromaticity, valence, and stereochemistry.
Finally, put the output SMILES in a json format: {"output": "predicted_valid_SMILES"}.

Model reasoning trace:
To derive the SMILES notation for the molecule described in the caption, let us proceed step by step:

Step 1: Base structure
The molecule is based on **1,4-benzoquinone**.

Step 2: Substituents
The caption specifies three substituents:
1. **Methoxy group** ('C[O]') at position **2**.
2. **Hydroxy group** ('[OH-]') at position **3**.
3. **Methyl group** ('[CH3]') at position **5**.
These substituents must be added to the base structure at the specified positions.

Step 3: Locant-mapped SMILES construction
To construct the SMILES, we modify the base structure by adding the substituents at the correct positions:
1. The **methoxy group** at position 2 is added to the carbon adjacent to one of the ketone groups.
2. The **hydroxy group** at position 3 is added to the next carbon in the ring.
3. The **methyl group** at position 5 is added to the carbon adjacent to the other ketone group.

The resulting SMILES notation is: `` COC1=C(C(=O)C(=C(C1O)C)O) ``

Step 4: Validation
We verify the SMILES for correctness:
1. **Ring closure**: The ring structure is intact (six-membered ring).
2. **Substituents**: Methoxy, hydroxy, and methyl groups are correctly placed at positions 2, 3, and 5, respectively.
3. **Aromaticity**: The molecule is a quinone with alternating double bonds.
4. **Valence**: All atoms satisfy their valence requirements.

The SMILES is valid.

Step 5: Output in JSON format
The final output is:
````json  
{"output": "COC1=C(C(=O)C(=C(C1O)C)O)"}```

One Gold SMILES: COC1=C(O)C(=O)C(C)=CC1=O  
One Predicted Output: COC1=CC(=O)C(O)=C(C)C1O  
Molecule Exact Match: False

Figure 11: Molecule generation example: chain-of-thought reasoning given canonicalized base scaffold.

Molecule Caption: The molecule is a member of the class of monohydroxy-1,4-benzoquinones that is 1,4-benzoquinone which is substituted by a methoxy, hydroxy and methyl group at positions 2,3 and 5, respectively. It is a mycotoxin isolated from *Aspergillus fumigatus*. It has a role as an *Aspergillus* metabolite and a mycotoxin. It derives from a 2,3-dihydroxy-5-methyl-1,4-benzoquinone.

Enriched molecule representation:

===

1,4-benzoquinone: C1(=O)C=CC(=O)C=C1

IUPAC-aligned SMILES with locants position (e.g., :6a] and :9]):

1,4-benzoquinone: [C:1]1(=O)[CH:2]=[CH:3][C:4](=O)[CH:5]=[CH:6]1

Locant valence and neighbor information:

```
{'locant': '[C:1]', 'position': '1', 'valence': 4, 'immediate neighbors': ['[CH:2]', '=O', '[CH:6]']},
{'locant': '[CH:2]', 'position': '2', 'valence': 4, 'immediate neighbors': ['[C:1]', '[CH:3]']},
{'locant': '[CH:3]', 'position': '3', 'valence': 4, 'immediate neighbors': ['[CH:2]', '[C:4]']},
{'locant': '[C:4]', 'position': '4', 'valence': 4, 'immediate neighbors': ['[CH:3]', '[CH:5]', '=O']},
{'locant': '[CH:5]', 'position': '5', 'valence': 4, 'immediate neighbors': ['[C:4]', '[CH:6]']},
{'locant': '[CH:6]', 'position': '6', 'valence': 4, 'immediate neighbors': ['[CH:5]', '[C:1]']}
```

Other SMILES for your reference:

2,3-dihydroxy-5-methyl-1,4-benzoquinone: OC=C(C(=C(C(=O)O)C)C)C=O

hydroxy (as substituent with the anchor point): \*O

methyl (as substituent with the anchor point): \*C

===

Model reasoning trace (Claude-Sonnet-4.5):

I need to construct a SMILES string for a 1,4-benzoquinone derivative with:

- Position 2: methoxy group
- Position 3: hydroxy group
- Position 5: methyl group

Let me work through this step by step using the locant information provided.

Starting from the 1,4-benzoquinone core:

[C:1]1(=O)[CH:2]=[CH:3][C:4](=O)[CH:5]=[CH:6]1

Now I need to add substituents:

- At position 2 ([CH:2] becomes [C:2]): add methoxy (OCH3 or COC)
- At position 3 ([CH:3] becomes [C:3]): add hydroxy (OH)
- At position 5 ([CH:5] becomes [C:5]): add methyl (CH3 or C)

Let me construct the SMILES:

Position 1: C(=O) – carbonyl carbon  
 Position 2: C with methoxy – C(OC)  
 Position 3: C with hydroxy – C(O)  
 Position 4: C(=O) – carbonyl carbon  
 Position 5: C with methyl – C(C)  
 Position 6: CH

Building the structure:

- C1(=O) at position 1
- C(OC) at position 2 (methoxy substituent)
- C(O) at position 3 (hydroxy substituent)
- C(=O) at position 4
- C(C) at position 5 (methyl substituent)
- CH at position 6

Connecting them in a ring:

C1(=O)C(OC)=C(O)C(=O)C(C)=C1

Let me verify:

- Position 1: C with =O and connected to positions 2 and 6 ✓
- Position 2: C with OC (methoxy) and double bond to position 3 ✓
- Position 3: C with O (hydroxy) and connected to position 4 ✓
- Position 4: C with =O and connected to positions 3 and 5 ✓
- Position 5: C with C (methyl) and double bond to position 6 ✓
- Position 6: C connected to positions 5 and 1 ✓

Alternative simpler notation:

COC1=C(O)C(=O)C(C)=CC1=O

{"output": "COC1=C(O)C(=O)C(C)=CC1=O"}

One Gold SMILES: COC1=C(O)C(=O)C(C)=CC1=O

One Predicted Output: COC1=C(O)C(=O)C(C)=CC1=O

Molecule Exact Match: True

Figure 12: Molecule generation example: chain-of-thought reasoning given enriched molecule representations and anchoring substituents.

After analyzing the GPT-4o failures, I've identified several critical issues in the chain-of-thought traces:

```

Common Failure Patterns

1. Misunderstanding ring numbering notation - Confusing SMILES ring closure digits (e.g., "1" in "s1cnccl1") with position locants
2. Incorrect substituent attachment - Adding substituents without properly accounting for which hydrogen/bond is being replaced
3. Breaking aromaticity rules - Creating structures where aromatic carbons have wrong valency after substitution
4. Ignoring provided reference SMILES - Not using the helpful reference structures as templates
5. Not verifying connectivity - Failing to trace through the complete structure to ensure proper bonding

Improved Protocol for GPT-4o
...
MOLECULE GENERATION FROM TEXT DESCRIPTION - IMPROVED PROTOCOL

You are an expert in computational chemistry. Your task is to generate a valid SMILES string from a textual molecule description.

CRITICAL RULES TO FOLLOW:

1. UNDERSTAND LOCANT POSITIONS VS RING CLOSURE NUMBERS
 - Locant positions (e.g., "position 5", "position 7") refer to atom numbering in the chemical structure
 - SMILES ring closure digits (e.g., "1" in "c1ccccc1") are just syntax for closing rings
 - These are DIFFERENT SYSTEMS - do not confuse them
 - Use the provided locant-mapped SMILES to identify which atom corresponds to which position

2. WORK WITH THE LOCANT-MAPPED SMILES
 - Start with the provided locant-mapped SMILES as your working template
 - Identify the EXACT atom at each position mentioned in the caption using the locant map
 - Check the "immediate neighbors" information to understand the local connectivity

3. ADD SUBSTITUENTS CORRECTLY
 - When adding a substituent at a position:
 a) Find the corresponding locant (e.g., [cH:7] for position 7)
 b) Note what type of atom it is (aromatic c, aliphatic C, etc.)
 c) Check its current neighbors and valence
 d) If it's [cH:X], you're replacing the implicit H with your substituent
 e) If it's [C:X], you're adding to an atom that may already have substituents
 - Remove the locant notation after modification but preserve atom type (aromatic vs aliphatic)

4. PRESERVE AROMATICITY
 - Aromatic carbons (lowercase c) must maintain exactly 3 bonds in the aromatic system
 - When you add a substituent to an aromatic carbon:
 * It replaces an implicit hydrogen
 * The carbon remains aromatic (lowercase c)
 * The carbon now has: 2 aromatic bonds + 1 substituent bond = 3 bonds total
 - NEVER create situations where aromatic carbons have 4 single bonds

5. USE REFERENCE SMILES AS TEMPLATES
 - If a related structure is provided (e.g., "2-oxido-5-methylquinone"), examine it carefully
 - These references often show the CORRECT way to attach similar substituents
 - Copy the attachment pattern from the reference when applicable

6. VERIFY YOUR STRUCTURE STEP-BY-STEP
 Before finalizing, check:
 - Count bonds for each atom to verify valence rules
 - Ensure all rings are properly closed
 - Confirm aromatic atoms (lowercase) only in aromatic systems
 - Verify aliphatic atoms (uppercase) where appropriate
 - Check that substituent attachment points make chemical sense

7. STEP-BY-STEP CONSTRUCTION PROCESS
 Step 1: Write out the locant-mapped SMILES
 Step 2: For each position mentioned in the caption:
 - Locate the atom using the locant notation
 - Note its current structure: [cH:7] or [C:5], etc.
 - Determine what's being replaced (usually an H)
 Step 3: For each substituent:
 - Write it out with its attachment point marked
 - Substitute it at the correct locant position
 - Preserve aromaticity (lowercase c for aromatic)
 Step 4: Remove all locant notation [X:Y] but keep atom types correct
 Step 5: Verify the structure makes chemical sense
 Step 6: Validate valence, rings, and aromaticity

8. EXAMPLE OF CORRECT SUBSTITUTION
 If adding a methyl group to position 5 on an aromatic ring:
 - Start: [c:5]1[cH:6]...[cH:4]
 - The [c:5] means aromatic carbon at position 5
 - After adding CH3: C(C)c1c... (aromatic c becomes aliphatic C if it gains sp3 character)
 - OR if staying aromatic: c1c(C)c... (aromatic c keeps lowercase, methyl in parentheses)
 - Check the reference SMILES to see which pattern is correct

9. WHEN YOU GET AN ERROR
 - If told "aromatic valency rule violated":
 * You likely made an aromatic carbon have 4 bonds
 * Re-count bonds at each aromatic position
 * Ensure substituents are in parentheses: c(SUBSTITUENT) not cSUBSTITUENT
 - Go back to the locant-mapped SMILES and rebuild carefully

10. FINAL OUTPUT
 Provide your answer as: {"output": "SMILES_STRING"}
 The SMILES must be valid, canonical, and chemically sensible.

WORK METHODICALLY. DO NOT RUSH. DOUBLE-CHECK AROMATICITY AND VALENCE AT EVERY STEP.
...

This protocol addresses the specific failures by emphasizing the distinction between locant positions and ring closures, requiring explicit valence checking, and providing clear rules for maintaining aromaticity during substitution.

```

Figure 13: Teacher-guided protocol example: failure pattern recognition of GPT-4o reasoning traces by Claude-Sonnet-4.5.