Molecule Generation through Reasoning with Large Language Models

Congfeng Cao[*,1], **Di Wu**[*,1],
Zhi Zhang[1], **Jia-Hong Huang**[1,2], **Jelke Bloem**[1], **Khalil Sima'an**[1],

[1]University of Amsterdam, [2]Amazon AGI,
[*]Equal contribution.
Correspondence: c.cao@uva.nl

Abstract

Molecule generation is significant for its potential in scientific discovery and practical applications. Recent attempts often frame this task as a *translation* problem from molecular caption to structural representation, such as SMILES. This paper first examines the feasibility of modeling the task as a reasoning process with large language models (LLMs), generating higher-quality molecules through structural decomposition and recombination within curated Chain-of-Thought (CoT). We then introduce a workflow for curating accurate CoT data, incorporating both machine and expert verification. Lastly, we demonstrate that with a limited dataset of 4,213 curated samples, namely **MolCoT4K**, we can elicit strong reasoning capabilities for molecule generation in open-source LLMs such as Qwen2.5-7B, achieving state-of-the-art exact match accuracy over strong open-source baselines, such as MolT5, LlaSMol, and ether0, as well as advanced commercial LLMs such as GPT-4o. Moreover, the resulting model, **MolGeneration**, attains a Pass@16 exact match accuracy of 48.46%, highlighting its strong potential for real-world experimental applications when supported by a feasible external verifier or chemistry experts. Our analysis shows that the correctness of the CoT path is crucial, while reasoning ability primarily enhances accuracy in fine-grained molecule generation.[1]

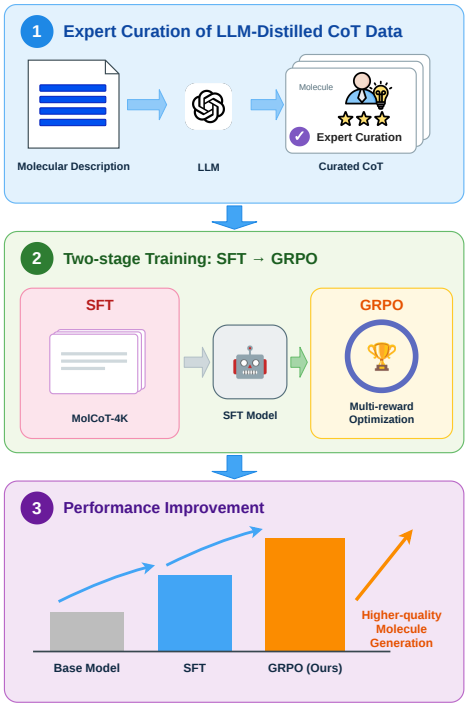


Figure 1: Overview of our molecule generation framework, where CoT data are distilled from an LLM and curated by experts to construct MolCoT-4K, followed by a two-stage post-training strategy consisting of SFT and GRPO with molecule-specific rewards, progressively improving molecule generation performance.

1 Introduction

Molecule generation plays a vital role in many aspects of modern life, such as accelerating the discovery of new materials and drugs by directly proposing a few candidate molecules instead of exhaustively searching through vast chemical spaces. This task can be framed as a mapping from molecular properties, functions, or reactions to structural representations, such as SMILES[2] (Weininger, 1988), a specification in the form of a short string for describing the structure of chemical species.

Given its substantial practical and scientific significance, considerable effort has been devoted to the problem of language model-based generation (Edwards et al., 2022; Pei et al., 2023; Liu et al., 2023; Yu et al., 2024)). In particular, to enable controllable and convenient generation of molecules with desired properties, Edwards et al.

[1]Dataset, model, and training codebase available at https://github.com/caocongfeng/MolGeneration.git

[2]https://en.wikipedia.org/wiki/Simplified_Molecular_Input_Line_Entry_System

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 3862–3879
November 6-10, 2026 ©2026 Association for Computational Linguistics

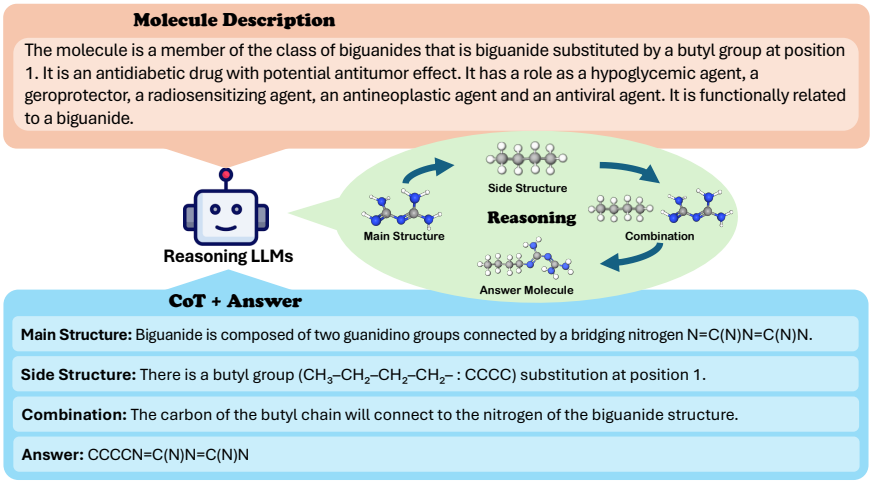


Figure 2: Illustration of molecule generation via reasoning. Given a molecular description, a reasoning LLM first decomposes the target molecule into a main scaffold N=C(N)N=C(N)N and a side chain CCCC, then attaches the side chain at position 1 to produce the final molecule CCCCN=C(N)N=C(N)N. See Appendix L Table 7 for additional examples.

(2022) first formulates this as a translation task-directly translating a natural language description of molecular properties into the SMILES structural representation. Yu et al. (2024) extends this paradigm to large language models (LLMs), achieving improved generation accuracy through fine-tuning the models on large-scale pairwise data. Further studies employ reinforcement learning and distilled reasoning data to improve generation performance, including ether0 (Narayanan et al., 2026), MolReasoner (Zhao et al., 2025), and Chem-R (Wang et al., 2026).

Although promising, several shortcomings remain: 1) these approaches heavily rely on the availability of large-scale data, which are particularly costly in the molecular domain and often requires extensive experimental validation and expert knowledge (Mayr et al., 2018); and 2) the compositional information of molecules and functional groups is not explicitly leveraged, but is instead left to black-box end-to-end training or training on distilled data without curation or verification. 3) distilled data contain various errors, even pertaining to basic chemical knowledge, such as molecular structures and the number of carbon bonds. Such errors are particularly critical in the chemistry domain but are not analyzed in these works.

In this paper, as shown in Figure 1, we investigate molecule generation by framing it as a step-by-step reasoning task. We first distill molecule generation CoT data from a LLM and have university-level experts correct the distilled CoT. We then apply a two-stage training strategy. As illustrated in Figure 2, the model is first prompted to explicitly generate intermediate structures and functional groups from the molecular description, such as the *main structure* and *side structure* (substituents), and then to construct the final structural representation (SMILES) based on these components. This compositional reasoning paradigm produces interpretable traces that align with chemical knowledge and improve generation accuracy.

To facilitate this paradigm, we construct the first curated chain-of-thought (CoT) dataset for molecule generation, **MolCoT-4K**. In particular, we employ GPT-4o-mini (OpenAI, 2024) to produce raw CoT data, and subsequently develop a curation tool that enables chemical experts to efficiently label, verify, and refine these outputs. As shown in Appendix E, our curation tool provides an interactive interface that enables experts to assign scores, make corrections, and record the rationale for the scoring. Our authors, with university-level knowledge of chemistry, label $4,213$ CoT examples for error types using the five-type score based on the labeling criteria (see Appendix F), and subsequently correct the errors. Our curation work differs from prior studies that directly distill molecule-related Chain-of-Thought (CoT) to build chemistry-related LLMs without human curation (Zhao et al., 2025; Narayanan et al., 2026; Zhuang et al., 2025). Our curation shows in Appendix H that the distilled CoT data contain incorrect descriptions that vary across molecular structures and properties,

and that only 8% of the distilled CoT data are correct. We further assess annotator agreement among the three independent university-level experts, observing 93.3% unanimous agreement, a mean Cohen's κ of 0.767, and Fleiss' κ of 0.738, indicating substantial agreement.

Building upon MolCoT-4K, we frame a two-step training strategy to equip open-source LLMs with molecule generation ability. Specifically, we use Qwen2.5-7B as the testbed and first perform supervised fine-tuning (SFT) on MolCoT-4K, which yields clear chemistry reasoning traces and substantial exact-match improvements over the base model. We then apply reinforcement learning (RL), where the model is encouraged to generate both a reasoning trace and a final molecule representation from the molecular description.

Finally, we integrate the SMolInstruct molecule generation subset with our curated CoT samples to develop our molecule generation model, **MolGeneration**. On the real-world evaluation dataset SMolInstruct, our MolGeneration outperforms both commercial LLMs and specialized molecular models (e.g., MolT5), achieving a state-of-the-art Exact Match (EM) score of 44.97%, surpassing GPT-4o (6.40%), Gemini-2.5 (19.49%), and MolT5 (31.70%). Moreover, assuming access to an external verifier, our model reaches a 48.46% EM score under 16-fold sampling, showing potential for real-world molecule generation when chemical experts are engaged in selecting candidates.

Our contributions are as follows:

- We demonstrate the effectiveness of framing molecule generation as a reasoning task: by deconstructing individual functional groups, generating intermediate results, and recombining them, one can trace a coherent reasoning path for molecule generation that leads to more accurate molecular outcomes.
- We present **MolCoT-4K**, a sample-efficient, curated reasoning dataset constructed with both machine and expert verification to maximize data availability, and release it to the community. To the best of our knowledge, MolCoT-4K is the first curated and agreement-checked CoT dataset for reasoning-based molecule generation.
- Based on MolCoT-4K, we provide our specialized molecular LLM, **MolGeneration**, achieving state-of-the-art performance in molecule generation, substantially surpassing current specialized models as well as advanced commercial LLMs. Furthermore, with access to an assumed verifier, the notable improvement highlights a strong potential in a real-world experimental environment.

2 Related Work

Molecule representation. Molecule representation is foundational to encoding chemical structures and functions when building chemical models. Traditional hand-crafted representations include RDK (Schneider et al., 2015), Morgan (Cereto-Massagué et al., 2015), and MACCS fingerprinting (Durant et al., 2002). RDK is more sensitive to structural changes compared to Morgan and MACCS fingerprinting (O'Boyle and Sayle, 2016).

In contrast, string-based representations such as the Simplified Molecular-Input Line-Entry System (SMILES) (Weininger, 1988) serialize molecular graphs into human-readable strings. SMILES are generated from molecular graphs using depth-first traversal. SMILES has become the standard for representing molecules in large chemical databases and machine learning pipelines, owing to its simplicity and compatibility with language models (Edwards et al., 2022, 2021; Yu et al., 2024). SELFIES (Self-Referencing Embedded Strings) (Krenn et al., 2020) was introduced as a robust molecular string representation. However, prior work (Yu et al., 2024) has shown that SELFIES offers no clear advantage for molecule generation, and the amount of annotated SELFIES data remains relatively limited.

In this study, we adopt SMILES as the molecular representation and apply RDKit (Landrum et al., 2025) to convert output into canonical SMILES (Weininger et al., 1989), a uniform SMILES representation, to address the potential many-to-one mapping issue for evaluation.

Language models with molecule generation. For convenient and controllable generation, Edwards et al. (2022) formulate molecule generation as a conditional language modeling task, translating natural language descriptions of molecular functions and properties directly into the corresponding SMILES. They post-train the multilingual language model T5 (Raffel et al., 2020) with molecular datasets, resulting in MolT5 for both molecule generation and captioning. BioT5 (Pei et al., 2023) and BioT5+ (Pei et al., 2024) introduce a comprehensive pre-training framework that en-

riches cross-modal integration in biology through chemical knowledge and natural language associations. Mol-Instructs (Fang et al., 2024) is designed to boost LLM proficiency in the specialized domain of bio-molecular studies. LlaSMol (Yu et al., 2024) shows that post-trained LLMs can achieve strong results on a comprehensive set of chemistry tasks, including molecule generation, outperforming commercial LLMs like GPT-4o (Achiam et al., 2023) and Claude 3 Opus by a significant margin. Furthermore, Jiang et al. (2024) propose a novel framework that interweaves model fine-tuning with data augmentation to mitigate the scarcity of high-quality data.

Zhuang et al. (2025) propose MPPReasoner, a multimodal large language model that incorporates generated chemical reasoning for molecular property prediction by distilling three LLMs. MolReasoner (Zhao et al., 2025) introduces a two-stage framework designed to transition LLMs from memorization to chemical reasoning via synthetic Chain-of-Thought (CoT) samples generated by GPT-4o (OpenAI, 2024). Narayanan et al. (2026) report Ether0, a large language model that can reason in natural language and respond with chemical structures, trained on distilled data from DeepSeek-R1 (DeepSeek-AI, 2025). Chem-R is trained through a three-phase framework, including Chemical Foundation Training, Chemical Reasoning Protocol Distillation, and Multi-task Group Relative Policy Optimization (Wang et al., 2026). All of these models generate molecules directly in a black-box or via distillation from strong LLMs, relying heavily on the availability of large-scale paired datasets. However, these distilled datasets contain various errors, which are particularly critical in the chemistry domain, and are not analyzed in these works.

Our work addresses this gap by eliciting step-by-step molecular reasoning abilities from LLMs via curated data, analyzing the influence of errors, and enabling molecule generation through structural decomposition into functional groups.

Datasets for molecule generation. ChEBI-20 (Edwards et al., 2021) is designed for molecule retrieval. It consists of 33,010 molecule-description pairs and serves as the post-training dataset for MolT5. ChEBI-20 contains descriptions of more than 20 words. LlaSMol introduced SMolInstruct, a large-scale, high-quality molecular dataset, which covers 14 selected chemistry tasks and 3.3 million samples, providing a foundation for training and evaluating LLMs in chemistry. The molecule generation part of SMolInstruct, which consists of 56,498 samples, is much larger than ChEBI-20. The L+M-24 dataset (Edwards et al., 2024) specifically targets molecule generation and captioning. Additionally, existing databases such as ZINC (Irwin and Shoichet, 2005), ChEMBL (Gaulton et al., 2012), and PubChem (Kim et al., 2021) offer molecule SMILES and description paired data. These datasets are large, but noisy.

In this paper, we introduce **MolCoT-4K**, a molecular reasoning dataset constructed through a combination of machine verification and expert review. It comprises 4,213 samples, each verified via exact matching and annotated into five levels of reasoning trace availability (ranging from 1 to 5; see Section 3.2). Human experts then refine each CoT to ensure chemical soundness, yielding the final gold-standard dataset. To the best of our knowledge, this is the first curated dataset explicitly designed to elicit molecular reasoning ability in LLMs.

3 Data Collection, Processing, and Curation

3.1 Data Collection and Filtering

We first collect description-molecule pairs from the train splits of three widely used datasets, specifically ChEBI-20, SMolInstruct, and L+M-24, resulting in a total number of 243,411 pairwise samples. Then, considering noisy and invalid data, we apply multiple steps of data filtering: 1) we remove samples when the corresponding SMILES cannot be processed by RDKit, including those that have invalid or incomplete molecular representations, or failure in 3D conformation generation; 2) we discard samples whose descriptions lack a clear molecular definition by requiring the text length to exceed 20 words[3], following the construction rule of ChEBI-20. In the end, we obtain a dataset D containing 55,236 samples, where each sample i is a tuple (m_i, s_i) corresponding to a molecule description and its SMILES representation.

3.2 Chain-of-Thought Data Construction

To maximize efficiency in data construction, given a pairwise molecule dataset D, we prompt GPT-

[3]For example, a short unclear description would be: "The molecule is cancer treatment." for $CN_1CCN(CC_2CCN(C_3C(N)=CNCC_3F)CC_2)CC_1$.

4o-mini k times for each molecule description (m), obtaining k corresponding potential CoT reasoning processes (c') prior to generating the final SMILES (s'). This procedure yields a raw dataset D' in which each sample i consists of k tuples $(m_i, c'_{ij}, s'_{ij}, s_i)$, where $j \in [1, k]$. Then, for each sample i, we retain it only if there exists an s'_{ij} that either exactly matches the gold answer s_i or can be converted into s_i using the RDKit tool. If multiple matches are found, we randomly select one s'_{ij} and the corresponding c'_{ij} as machine-generated c'_i and s'_i. This filtering process substantially reduces the dataset size from 55,236 to 4,213 samples, reflecting the difficulty current LLMs face in handling this task effectively. In the resulting dataset D'', each sample is a tuple (m_i, c'_i, s'_i).

Note that both c'_i and s'_i are machine-generated. While s'_i aligns exactly with the gold answer s_i, the availability of c'_i is not guaranteed. In practice, we observe that a considerable proportion of c' lacks sound chemical inference, e.g., main structure errors; see Appendix H for the error distribution within the dataset. To maximize data availability, we involve human experts to manually revise each c'_i for correctness based on manually assigned error labels (see Appendix F for labeling criteria). We further develop a CoT curation tool that supports error-level annotation, manual correction, and related I/O functions to accelerate the process; see Appendix E for details.

Finally, we construct MolCoT-4K, denoted as D_{MolCoT}, an expert-curated molecule generation dataset of 4,213 samples, split into train/valid/test sets in an 8:1:1 ratio. Each sample is a tuple (m_i, c^*_i, c'_i, s'_i), where c^*_i is a CoT trace with chemical validity verified by experts. To simplify, we use (m_i, c_i, s_i) to denote (m_i, c^*_i, s'_i) for samples in D_{MolCoT} in the following text, unless specified otherwise.

3.3 Expert Agreement Check

To evaluate the reliability of our expert curation, we randomly sample an agreement subset $D_{\text{agr}} \subset D_{\text{MolCoT}}$ containing 30 samples from MolCoT-4K (D_{MolCoT}). The mean CoT length of the randomly sampled instances is 1,115 characters, closely matching the overall mean length of 1,185 characters in MolCoT-4K.

We observe 93.3% unanimous agreement (28/30) across the three experts. Individually, Expert 1 labels 28/30 instances as correct, while Expert 2 and Expert 3 each label 29/30 instances as correct. The pairwise Cohen's κ values are 0.651 (Expert 1-Expert 2), 0.651 (Expert 1-Expert 3), and 1.000 (Expert 2-Expert 3), with a mean κ of 0.767. The overall multi-rater agreement measured by Fleiss' κ is 0.738, which also falls into the agreement range. These results provide consistent evidence that expert verification of molecular reasoning traces is reliable and reproducible.

4 Methods

This section outlines the training approach used to elicit chemical reasoning ability in LLMs. We adopt a two-stage post-training strategy. First, we apply SFT using (m_i, c_i, s_i) in MolCoT-4K as a warm-up stage to equip the model with the basic ability to generate CoT responses. Note that the model is supervised by c and s given a description m. We then apply reinforcement learning to further improve the model's reasoning capability.

4.1 Reinforcement Learning with Molecule Reward

We apply the widely used GRPO (Shao et al., 2024) reinforcement learning method, to further improve the model's reasoning ability after SFT. GRPO discards the value function and estimates the advantage in a group-wise normalization. GRPO typically introduces a KL penalty term to regularize divergence between the online and reference policies, while we eliminate the KL penalty term to allow the policy to explore more aggressively in the chemical space. The formal derivation can be found in Appendix O.

4.2 Molecular Reward Design

We follow RL applications to math reasoning (Shao et al., 2024; Yu et al., 2026; Liu et al., 2025) and design rewards based on correctness and output format constraints. Similar to mathematical reasoning, evaluating the correctness of molecular reasoning relies on structural and chemical consistency, providing clear criteria for reward design. The correctness reward ensures that the model receives a reward only when the generated SMILES exactly matches the ground-truth SMILES. Our preliminary experiments show minimal gains when using similarity-based rewards, e.g., those from the RDKit tool (Landrum et al., 2025). Thus, we chose the simplest binary reward version. The format reward ensures that the model receives a reward only when the generated CoT contains both the `<Think>` and `<Answer>` tags. However, for molecule generation,

correctness and formatting alone are insufficient due to the complexity of chemical syntax and combinations. We therefore design a molecule-specific ensemble reward strategy, where we additionally introduce a SMILES validity reward that ensures the generated output corresponds to a chemically valid molecule, verified by the RDKit tool.

5 Experimental Setup

Base models. We conduct experiments on recent open-source LLMs of the Qwen (Qwen, 2025a,b), LLaMA (Grattafiori et al., 2024), and Mistral (Jiang et al., 2023) families, covering model sizes from 0.5B to 32B.

Evaluation metrics. We employ three evaluation metrics ranging from coarse to fine: Validity, RDK-FTS (Schneider et al., 2015), and Exact Match (EM). **Validity** measures the proportion of generated SMILES strings that correspond to chemically valid molecules, regardless of correctness, processed by the RDKit tool. **RDK-FTS** encodes molecules using RDKit fingerprints and calculates the mean Tanimoto similarity (also known as the Jaccard index) between the generated and reference SMILES strings. Finally, **EM** requires the generated output to be identical to the reference SMILES, or convertible to the exact same SMILES using the RDKit tool.

Training setting. Our experiments were conducted with Transformers (Wolf et al., 2020) and TRL (von Werra et al., 2020) on NVIDIA H100 (100 GB) GPUs. In the SFT stage, we train for three epochs with a learning rate of 5×10^{-5}. We set the maximum sequence length to 2,048 tokens with AdamW, using a 0.03 warmup. In the RL stage, we apply GRPO for one epoch with a learning rate of 1×10^{-6}. We set the maximum sequence length to 2,048 tokens with AdamW, using a 0.1 cosine warmup. Training details can be seen in Appendix J for SFT and Appendix K for RL.

Decoding Strategy. We employ two decoding strategies to evaluate the effectiveness of our method and the resulting model. (1) Beam search is applied to demonstrate performance in a constrained experimental setting; and (2) Best-of-N sampling is applied to approximate the upper bound of performance (Pass@N) when an external verifier is available. For Best-of-N settings, we set top-p as 0.9, top-k as 50, and temperature as 1.0, to ensure that samples are drawn from relatively higher-probability regions while maintaining diversity.

6 Experimental Results and Analysis

6.1 On the Role of Reasoning for Molecule Generation

In this section, we begin by evaluating the impact of incorporating decomposition and recombination information, represented as CoT traces, prior to generating the final SMILES strings. Experiments are conducted on the MolCoT-4K dataset (see Section 3.2) across a range of LLMs differing in size and family. We focus here on the simplest setting, supervised fine-tuning, where w/ CoT and w/o CoT denote training with (m, c, s) and (m, s) pairs, respectively, from MolCoT-4K.

Figure 3 shows the corresponding results on Qwen2.5 and Qwen3 families, measured at progressively finer levels of granularity by Validity, RDK-FTS, and EM scores (see Section 5). We observe that: (1) Within the Qwen family (e.g., Qwen2.5 from 0.5B to 32B), performance improves steadily as the model size increases; (2) Compared to the base models, supervised fine-tuning substantially improves molecule generation performance. Moreover, incorporating CoT data during training (w/ CoT) consistently outperforms the traditional translation setting (w/o CoT), underscoring the necessity of explicitly decomposing and recombining structural information prior to generating the final answer; (3) Our reasoning model shows increasingly pronounced benefits as the evaluation becomes more fine-grained. For example, although the traditional translation approach (w/o CoT) can also generate a comparable proportion of valid molecules, when molecular similarity, and especially the degree of exact matching, is considered, the superiority of reasoning models (w/ CoT) becomes more apparent. This demonstrates that reasoning ability plays a crucial role in fine-grained molecule generation, consistent with the chemical intuition that component-wise decomposition and recombination enable more precise generation. More results on the LLaMA and Mistral families (LLaMA-3.1-8B, Mistral-7B-v0.1, and Mistral-7B-v0.3) can be found in Table 1, showing that supervised fine-tuning with curated CoT data consistently improves the performance compared with both the base models and models trained without CoT.

We further investigate the effectiveness of in-

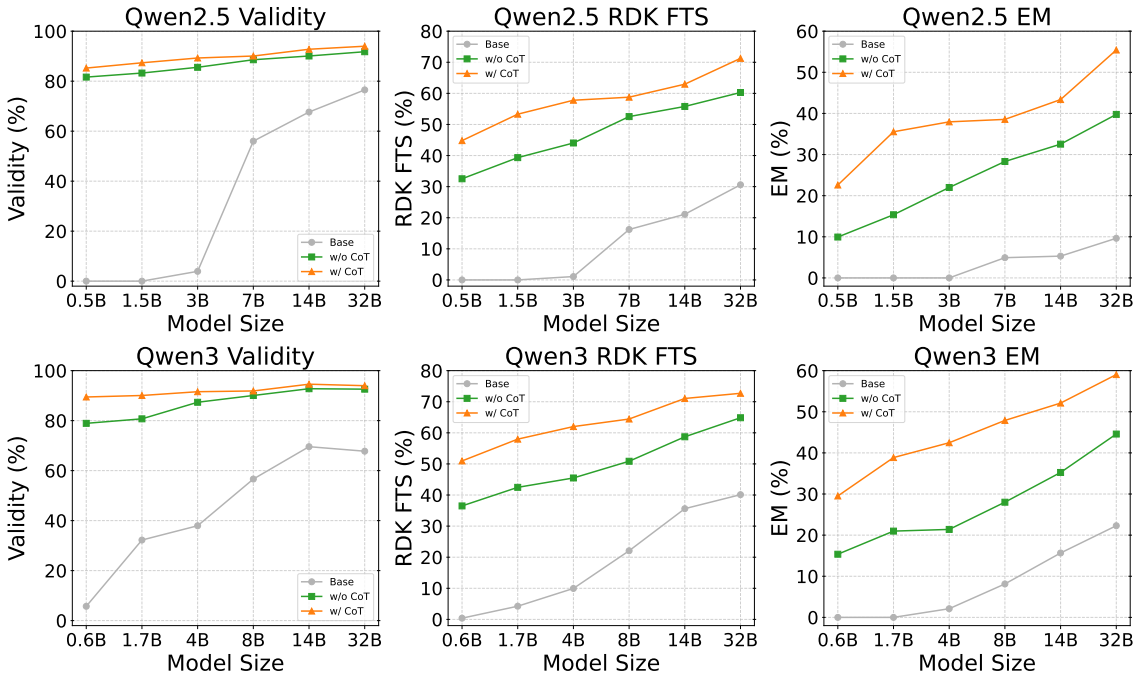


Figure 3: Performance comparison on our MolCoT-4K dataset across metrics, model scales, and training methods for Qwen2.5 and Qwen3. *Validity*, *RDK-FTS*, and *EM* represent average performance at progressively finer levels of granularity: the proportion of valid molecules, average structural similarity, and exact match accuracy, respectively. The results show that reasoning ability contributes most to improvements in fine-grained molecule generation (EM). Detailed results, along with those for other models such as LLaMA and Mistral, are provided in Table 1.

Model	EM	RDK FTS	Validity
LlaMA-8B	0.00%	0.39%	5.72%
LlaMA-8B$_{\text{w/o CoT}}$	24.40%	45.98%	83.43%
LlaMA-8B$_{\text{w/ CoT}}$	33.74%	53.52%	87.65%
Mistral-7B-v0.1	0.00%	0.00%	0.00%
Mistral-7B-v0.1$_{\text{w/o CoT}}$	1.51%	9.43%	35.24%
Mistral-7B-v0.1$_{\text{w/ CoT}}$	4.21%	26.15%	83.43%
Mistral-7B-v0.3	1.50%	13.32%	52.71%
Mistral-7B-v0.3$_{\text{w/o CoT}}$	25.30%	44.72%	75.30%
Mistral-7B-v0.3$_{\text{w/ CoT}}$	34.34%	51.97%	87.05%

Table 1: Performance comparison on the MolCoT-4K across LlaMA and Mistral families. Results are reported for three settings: raw base models without fine-tuning, models fine-tuned without CoT data, and models fine-tuned with CoT data. The comparison demonstrates that CoT consistently improves performance across architectures and scales.

Model	EM	RDK-FTS	Validity
Qwen2.5-7B	4.92%	16.22%	56.00%
+ SFT$_{\text{w/o CoT}}$	28.31%	52.53%	88.57%
+ SFT$_{\text{w/ CoT}}$	38.55%	58.77%	90.06%
+ RL + SFT$_{\text{w/o CoT}}$	20.92%	37.75%	73.85%
+ RL + SFT$_{\text{w/ CoT}}$	**43.98%**	**61.63%**	**91.27%**
$\Delta_{\text{Improvement}}$	+39.06%	+45.41%	+35.27%

Table 2: Performance comparison under different training strategies, measured by Exact Match (EM), RDK-FTS, and validity scores, respectively.

corporating both SFT and RL for molecule generation, i.e., the two-stage post-training strategy, see Appendix O. Experiments are also conducted on the MolCoT-4K dataset using a series of ablation strategies: (1) SFT using only pairwise data (m, s) and discarding CoT, denoted as **SFT$_{\text{w/o CoT}}$**, following the traditional translation paradigm (Edwards et al., 2021; Yu et al., 2024); (2) SFT with CoT data, i.e., (m, c, s), denoted as **SFT$_{\text{w/ CoT}}$**; (3) We additionally incorporate RL training stage after (1) and (2) to show further impacts, denoted as **RL+SFT$_{\text{w/o CoT}}$** and **RL+SFT$_{\text{w/ CoT}}$**, respectively.

Table 2 shows the results for these settings on the Qwen2.5-7B model, measured by EM, RDK-FTS, and Validity scores. The results clearly indicate that the base model (Qwen2.5-7B) performs poorly on this task, achieving only 4.92% EM, 16.22% RDK-FTS, and 56.00% Validity scores. Supervised fine-tuning with pairwise data (SFT$_{\text{w/o CoT}}$), i.e., translation modeling, substantially improves performance, particularly by increasing the validity of generated SMILES strings and enhancing structural similarity. When incorporating CoT data (SFT$_{\text{w/ CoT}}$) for

Model	EM	RDK-FTS	Validity
Qwen2.5-7B	4.92%	16.22%	56.00%
$SFT_{w/ corrected CoT}$	**20.48%**	**43.22%**	**71.45%**
$SFT_{w/ minor issue}$	17.77%	42.96%	69.58%
$SFT_{w/ element issue}$	16.27%	41.29%	64.52%
$SFT_{w/ structural issue}$	15.36%	40.13%	64.34%
$SFT_{w/ ineffective CoT}$	7.53%	30.26%	58.73%
$SFT_{w/ messed-up CoT}$	0.60%	4.95%	20.19%

Table 3: The effectiveness of six groups of data of varying noise type within CoT across EM, RDK-FTS, and Validity. Corrected CoT achieves the best performance as expected. Performance gradually decreases as the noise increases.

Model	EM	RDK FTS	Validity
Gemini-2.5-pro	19.49%	62.72%	81.18%
Claude 3 Opus	12.30%	57.60%	92.60%
GPT-4o	6.40%	42.60%	81.40%
Gemini-2.0-flash	5.29%	27.44%	64.38%
GPT-4o-mini	1.40%	12.63%	52.27%
MolT5	31.70%	**73.20%**	95.30%
Chem-R	29.76%	57.76%	86.40%
ether0	20.42%	39.43%	70.76%
LlaSMol	19.20%	61.70%	**99.70%**
MolInst	6.00%	43.60%	84.80%
ChemLLM	0.90%	14.30%	4.30%
MolReasoner	0.64%	6.47%	58.92%
Qwen2.5-7B	0.00%	1.08%	8.95%
$MolGeneration_{w/o CoT}$	37.02%	61.63%	89.81%
$MolGeneration_{w/ CoT}$	**44.97%**	67.69%	94.02%

Table 4: Performance comparison on the SMolInstruct dataset. Our model achieves the highest EM score of 44.97% using beam search (beam size = 5), outperforming both open-source LLMs (e.g., MolT5, Chem-R) and commercial LLMs, e.g., GPT-4o.

training, we observe clear reasoning trails during generation. This enhances performance, increasing exact matching results from 28.31% to 38.55% and highlighting a finding consistent with that in Figure 3 on the role in fine-grained generation. Further incorporating RL for training achieves the best results with 43.98% EM, 61.63% RDK-FTS, and 91.27% Validity scores. However, discarding CoT data (RL+$SFT_{w/o CoT}$) in this setting largely hurts performance, even worse than that of solely supervised fine-tuning ($SFT_{w/o CoT}$). The labels of the test dataset after SFT and RL are shown in Appendix I. Examples of results after RL are shown in Appendix L.

6.2 On the Role of Correctness of Reasoning Trace

To further examine how the correctness of the reasoning trace influences training in molecule generation, we conduct controlled experiments. From our CoT dataset, MolCoT-4K, where each sample is expressed as (m, c, s), we randomly select 1,000 samples and manually inject five types of noise into c. These noise types span mild to severe perturbations: CoT with minor issues, element issues, structural issues, ineffective CoT (i.e., removal of CoT), and a "messed-up" CoT created by shuffling the words within each reasoning path to maximize disruption (see Appendix G for details). This process yields six datasets in total, including the clean subset from MolCoT-4K. All six datasets share the same 1,000 molecular descriptions m and corresponding SMILES s, differing only in their CoT c for control across groups.

We also perform supervised fine-tuning on the Qwen2.5-7B model with each of these datasets. The results are presented in Table 3. Training with the dataset containing correct CoT yields the strongest performance, while increasing perturbation levels lead to a gradual decline. In particular, training with the "messed-up CoT" setting results in nearly collapsed performance. Using the ineffective CoT data (empty CoT) provides only marginal improvement over the base model, but still falls far short of the correct CoT setting, consistent with the observations in Section 6.1. Overall, the correctness of CoT is crucial to successfully elicit reasoning ability for molecule generation.

6.3 Experimental Results on SMolInstruct

In this section, we evaluate the effectiveness of incorporating reasoning ability on the widely used molecule generation benchmark, SMolInstruct. This dataset includes 56,498 description–SMILES pairs in the training set, which has generally been used for translation modeling (Yang et al., 2024). The test set comprises 2,493 samples. Even advanced commercial LLMs struggle on this task; for instance, GPT-4o achieves only a 6.4% EM score, as reported by Yu et al. (2024), posing a challenging benchmark.

As discussed in Section 6.1, traditional translation modeling struggles to capture fine-grained structural information, yet it can still generate SMILES with relatively high coarse-grained similarity and validity. To maximize generation performance, we combine the SMolInstruct training set with MolCoT-4K. Note that MolCoT-4K is constructed exclusively from the training splits of the source datasets and therefore contains no samples from the SMolInstruct test set used for evaluation.

To highlight the role of CoT data, we follow the setting in Figure 3 and apply supervised fine-tuning on the Qwen2.5-7B model using the merged dataset with and without the CoT path, yielding our **MolGeneration** models, denoted as $\text{MolGeneration}_{\text{w/ CoT}}$ and $\text{MolGeneration}_{\text{w/o CoT}}$. We conduct an extensive comparison with both open-source chemistry-specific systems and commercial LLMs. To ensure a fair comparison, we use beam search with a beam size of 5 as the decoding strategy, which is also applied on the baseline open-source systems, e.g., MolT5 and LlaSMol.

As shown in Table 4, our models achieve state-of-the-art performance in exact match, reaching 44.97% and substantially outperforming all baseline systems. The small scale of the CoT dataset contributes substantially, where training with CoT ($\text{MolGeneration}_{\text{w/ CoT}}$) clearly surpasses its translation-based version ($\text{MolGeneration}_{\text{w/o CoT}}$). Recent strong models, such as MolT5 and LlaSMol, perform considerably worse in EM (31.7% and 19.2%), although they attain the highest scores in structural similarity (73.20%) and validity (99.70%), respectively. Reasoning-based models trained with directly distilled CoT data, ChemR and ether0, achieve EM scores of 29.76% and 20.42%, respectively. This is consistent with our findings in Section 6.1, where reasoning primarily benefits fine-grained generation, while translation modeling mainly contributes at the coarse level.

Notably, Qwen2.5-7B, the base model we trained on, fails completely on this challenging dataset, producing no exact matching SMILES. GPT-4o-mini, the source from which our MolCoT-4K data was distilled, achieves only 1.4% in this benchmark. This underscores both the difficulty of the benchmark and the critical role of training data.

In addition, to explore the potential of MolGeneration for molecular design, we apply Best-of-N sampling to generate a set of N ($N = 1,2,4,8,16$) candidate molecules from MolGeneration. Assuming access to an external verifier, chemical experts can then select results from this candidate set. The Pass@N results are presented in Figure 4, where we observe a nearly linear trend as sample size N increases.

MolGeneration exhibits steady improvements from Pass@1 to Pass@16 and achieves a Pass@16 exact match score of 48.46%. Moreover, it consistently outperforms commercial LLMs such as Gemini-2.0-flash and GPT-4o-mini in this setting, while Qwen2.5-7B yields nearly 0.0% EM across all values of N. In short, MolGeneration demonstrates strong potential for future molecular design in real-world experimental settings.

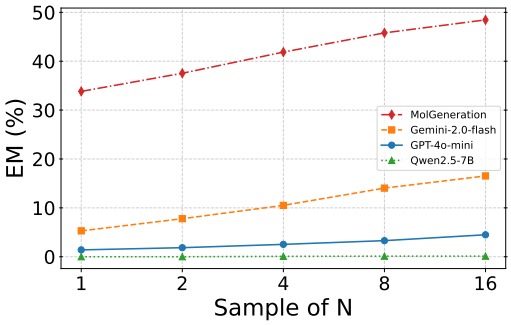


Figure 4: Performance comparison of Exact Match (EM) on CoT-augmented SMolInstruct across Pass@N ($N = 1,2,4,8,16$). MolGeneration exhibits steady improvements from Pass@1 to Pass@16, consistently outperforming commercial LLMs such as Gemini-2.0-flash and achieves a Pass@16 exact match score of 48.46%.

7 Conclusion

In this work, we frame molecule generation as a step-by-step reasoning process in which LLMs produce chemically valid CoT paths that decompose and recombine functional groups before generating the corresponding SMILES. To support this ability, we construct the first curated CoT dataset, **MolCoT-4K**, incorporating both machine and expert verification to maximize data reliability. Our extensive experiments demonstrate that (1) despite its small scale, MolCoT-4K can effectively elicit reasoning ability for this purpose; (2) the resulting reasoning models consistently outperform traditional translation-based models when trained on the same amount of data; and (3) the correctness of CoT is crucial to successfully elicit reasoning ability for molecule generation. Lastly, we present MolGeneration, which achieves state-of-the-art generation performance, surpassing both strong domain-specific and commercial LLMs, demonstrating strong potential for future molecular design.

8 Limitations

Our experiments demonstrate that MolCoT-4K can effectively elicit reasoning abilities for molecule generation even at a limited scale. However, curating this dataset required two months of effort from our team, despite the use of a dedicated curation tool to facilitate the process, highlighting the strong

dependence on domain knowledge and expert involvement. These limitations restrict the scale and scope of our approach to generating accurate CoT data in the chemistry domain.

As mentioned in Section 3 and Appendix F, the distilled CoT data contains various issues ranging from molecular structures to properties. However, some of these errors, such as incorrect molecule counts and inaccurate property descriptions, can be curated using automatic tools, such as agents that search the Internet or tools that perform automatic correction via LLMs. The lack of such tools limits our ability to curate large-scale, high-quality datasets, which we leave as a direction for future work.

9 Ethics Statement

Data sources and licensing. The datasets used in this work, including ChEBI-20, SMolInstruct, and L+M-24, are publicly available. The toolkits employed, including TRL and Transformers, are open-source and released under the Apache-2.0 License.

Potential risks. Our model aims to generate molecules from natural language descriptions via step-by-step reasoning. While our goal is to support chemistry research and development, such as drug discovery, the model could potentially be misused to generate harmful molecules. To mitigate these risks, we will release our model with explicit usage restrictions, limiting its application to legitimate chemistry research and prohibiting harmful or malicious use.

10 Acknowledgments

Correcting each sample is highly costly and requires domain expertise. Conducting the inter-annotator agreement check is also costly and time-consuming, even with our curation tools. We therefore sincerely thank Pengyu Zhang and Yuting Wang for volunteering as expert annotators for the inter-annotator agreement check.

References

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 arXiv:2303.08774*.

Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. 2015. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Preprint*, arXiv:2501.12948.

Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. 2002. Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Carl Edwards, Qingyun Wang, Lawrence Zhao, and Heng Ji. 2024. L+M-24: Building a dataset for Language+Molecules @ ACL 2024. In *Proceedings of the 1st Workshop on Language + Molecules (L+M 2024)*, pages 1–9, Bangkok, Thailand. Association for Computational Linguistics.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2024. Mol-Instructions: A large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*.

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and 1 others. 2012. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

John J Irwin and Brian K Shoichet. 2005. ZINC - a free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, 45(1):177–182.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego

de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *Preprint*, arXiv:2310.06825.

Yinuo Jiang, Xiang Zhuang, Keyan Ding, Qiang Zhang, and Huajun Chen. 2024. Enhancing cross text-molecule learning by self-augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9551–9565, Bangkok, Thailand. Association for Computational Linguistics.

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. 2021. PubChem in 2021: new data content and improved web interfaces. *Nucleic acids research*, 49(D1):D1388–D1395.

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.

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, NadineSchneider, Eisuke Kawashima, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, tadhurst cdd, Alain Vaucher, and 11 others. 2025. rdkit/rdkit: 2024_09_5 (q3 2024) release.

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

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding R1-Zero-like training: A critical perspective. In *Second Conference on Language Modeling*.

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. 2018. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical science*, 9(24):5441–5451.

Siddharth Narayanan, James Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Michael Pieler, Sam Rodriques, and Andrew White. 2026. Training a scientific reasoning model for chemistry. *Advances in Neural Information Processing Systems*, 38:157671–157710.

OpenAI. 2024. GPT-4o system card. *Preprint*, arXiv:2410.21276.

Noel M O'Boyle and Roger A Sayle. 2016. Comparing structural fingerprints using a literature-based similarity benchmark. *Journal of cheminformatics*, 8(1):36.

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1216–1240, Bangkok, Thailand. Association for Computational Linguistics.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. 2023. BioT5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1102–1123, Singapore. Association for Computational Linguistics.

Qwen. 2025a. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qwen. 2025b. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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

Nadine Schneider, Roger A Sayle, and Gregory A Landrum. 2015. Get your atoms in order– an open-source implementation of a novel and robust molecular canonicalization algorithm. *Journal of chemical information and modeling*, 55(10):2111–2120.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Weida Wang, Benteng Chen, Di Zhang, Wanhao Liu, Ben Gao, Shuchen Pu, Shuzhou Sun, Jin Zeng, Tianshu Yu, Wanli Ouyang, and 1 others. 2026. Chem-R: Learning to reason as a chemist. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 12291–12302.

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.

David Weininger, Arthur Weininger, and Joseph L Weininger. 1989. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences*, 29(2):97–101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. LlaSMol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. In *First Conference on Language Modeling*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Guojiang Zhao, Sihang Li, Zixiang Lu, Zheng Cheng, Haitao Lin, Lirong Wu, Hanchen Xia, Hengxing Cai, Wentao Guo, Hongshuai Wang, Mingjun Xu, Siyu Zhu, Guolin Ke, Linfeng Zhang, and Zhifeng Gao. 2025. MolReasoner: Toward effective and interpretable reasoning for molecular LLMs. *Preprint*, arXiv:2508.02066.

Jiaxi Zhuang, Yaorui Shi, Jue Hou, Yunong He, Mingwei Ye, Mingjun Xu, Yuming Su, Linfeng Zhang, Ying Qian, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2025. Reasoning-enhanced large language models for molecular property prediction. *Preprint*, arXiv:2510.10248.

A The Use of Large Language Models

In this paper, we prompt GPT-4o-mini to generate draft CoT traces from molecule descriptions prior to human post-editing to enable relatively large-scale data curation. In addition, we also evaluate Claude, GPT, and Gemini on the MolCoT-4K and SMolInstruct test sets for comparison.

B Distillation Prompt

The prompt used to extract Chain-of-Thought (CoT) data from GPT-4o-mini is shown in Figure 5. The {Molecule Description} is replaced by actual data samples.

C Commercial Reasoning Model API Pricing

We select GPT-4o-mini because it is one of the strongest-performing and most cost-effective models.

D Example of Dataset

Table 6 presents an example from the constructed dataset, which includes a molecule description of Glyphosate, its corresponding SMILES representation, and the distilled Chain-of-Thought (CoT) reasoning steps. The CoT first identifies the core structure of Glyphosate based on the description, then identifies components of this molecule, including glycine, the phosphonomethyl group, and N-alkylation, and identifies functional groups. Finally, it constructs the SMILES representation based on the above decomposition.

E Curation Tool

The curation is shown in Figure 6. The original CoT data is presented in the left CoT text box and can be edited. At the same time, we can label the data in the score text box and fill in the detailed issue in the reason text box. In this way, we manually correct and label the data. After editing, we can save the revision by clicking the Save changes button. The two text boxes above, which cannot be edited, are used to show the SMILES and molecule description, respectively. After saving, the curated result can be seen on the right side, including the SMILES, molecule description, corrected CoT, label, and issue. In addition, the modification record can be exported and viewed by clicking the view export button. The curation tool is available in the accompanying repository for this paper[4].

F Labeling Criteria

We first identify the different types of CoT (error types) and then correct CoTs that have errors. To identify the different types of CoT, we label the data according to the following criteria:

[4] https://github.com/caocongfeng/MolGeneration.git

User: You are an expert in info-chemistry, especially in SMILES interpretation and translation. Provide the SMILES representation based on the following molecule description: {Molecule Description}. Analyze it step by step, and return only the SMILES within the <Answer></Answer> tags.

Figure 5: Distillation prompt. Distill Chain-of-Thought (CoT) for molecule generation based on the above prompt from GPT-4o. The placeholder *{Molecule Description}* is filled with a molecule description.

Provider	Model	Input ($/1M tok)	Output ($/1M tok)
OpenAI	GPT-4o-mini	0.15	0.60
OpenAI	GPT-4o	2.50	10.00
OpenAI	o3	10.00	40.00
Anthropic	Claude Sonnet 3.7	3.00	15.00
Google	Gemini 2.5 Pro (Thinking)	1.25 ($\leq 200k$) / 2.50 ($\geq 200k$)	10.00 ($\leq 200k$) / 15.00 ($\geq 200k$)
Google	Gemini 2.5 Flash (Hybrid reasoning)	0.30	2.50
DeepSeek	DeepSeek-R1	0.14	2.19

Table 5: Commercial model API pricing ($/million tokens). GPT-4o-mini is one of the strongest-performing and most cost-effective models.

- **1 Ineffective CoT:** There is an ineffective reasoning path in the CoT, including repeated descriptions, empty content in thinking tags, and simple repetition of properties from the description. For example, the CoT for reasoning CC(=O)O[Sn](C$_1$=CC=CC=C$_1$)(C$_1$=CC=CC=C$_1$)C$_1$=CC=CC=C$_1$ just repeats its description: "The molecule is an organotin compound that is the O-acetyl derivative of triphenyltin hydroxide. A fungicide used to control blights on potatoes, leaf spot diseases on sugar beet and anthracnose on beans. It has a role as an antifungal agrochemical. It is an organotin compound and an acetate ester. It derives from a fentin hydroxide.".

- **2 Main-structure issue:** The CoT contains issues with the main structure, such as mistakenly identifying a linear structure as a branched or ring structure, or vice versa. For example, O=C(/C=C/C$_1$=CC=CC=C$_1$)OCC$_1$=CC=CC=C$_1$ contains ring structures (C$_1$CCCCC$_1$), whereas the reasoning path identifies it as linear structures (CCCCCC).

- **3 Element issue:** The CoT contains issues with element counts, such as reasoning a twelve-carbon linear chain as a ten-carbon linear chain.

- **4 Minor issue:** The CoT contains issues that may not affect the main structure, including missing chiral bonds and bonds lacking between two atoms. For example, it reasons C(=O)O (the carboxyl / carboxylic acid group, $-$COOH) as OCO, which lacks a bond between carbon and oxygen. Another issue is, for example, reasoning a Kekulé format SMILES into an aromatic format.

- **5 Correct CoT:** The CoT does not need any revision. An example can be seen in Table 6.

G MolCoT-1K Construction

To verify the role of correctness in the reasoning trace, we construct a dataset named **MolCoT-1K**.

We randomly select 1,000 samples from the MolCoT-4K training set and inject five types of CoT issues into c^*, respectively, resulting in $c^{*'}$. The five types of CoT issues are: CoT with minor issues, CoT with element issues, CoT with structural issues, ineffective CoT, and messed-up CoT. Specifically, for ineffective CoT, we remove the reasoning content to provide the answer directly. To inject main-structure issues, we modify linear chains into rings or branches, and vice versa. The modified structures are then used to alter the corresponding content in the CoT. For example, we replace *"linear carbon chain with six carbons, CCCCCC* with *"Benzene, a ring of six carbons, C$_1$=CC=CC=C$_1$"*. To inject element issues, we randomly add or remove carbon atoms, while for minor issues, we add or remove bonds between two atoms. In addition, to verify the effectiveness

Example Data: Glyphosate ($C_3H_8NO_5P$)

Molecule Description: The molecule is a phosphonic acid resulting from the formal oxidative coupling of the methyl group of methylphosphonic acid with the amino group of glycine. It is one of the most commonly used herbicides worldwide, and the only one to target the enzyme 5-enolpyruvyl-3-shikimate phosphate synthase (EPSPS). It has a role as an agrochemical, an EC 2.5.1.19 (3-phosphoshikimate 1-carboxyvinyltransferase) inhibitor and a herbicide. It is a phosphonic acid and a glycine derivative. It is a conjugate acid of a glyphosate(2-) and a glyphosate(1-).

CoT:

To generate the SMILES representation for the molecule based on the provided description, let's break down the components step by step.

1. **Identify the core structure**: The molecule in question is glyphosate, which is a derivative of glycine and also contains a **phosphonic acid** group.

2. **Components**:
   - **Glycine**: The simplest amino acid with the structure NH_2-CH_2-COOH.
   - **Phosphonomethyl group**: The substituent -CH_2-P(=O)$(OH)_2$ (a phosphonic acid attached via a methylene).
   - **N−substitution (N−alkylation)**: The -CH_2-P(=O)$(OH)_2$ group is bonded to the **amino nitrogen of glycine**, forming N−(phosphonomethyl)glycine.

3. **Determine functional groups**:
   - Glyphosate includes a **carboxylic acid** (-COOH), a **phosphonic acid** (-P(=O)$(OH)_2$), and a **secondary amine** (-NH-).
   - The overall structure has the amine nitrogen linked via a methylene (-CH_2-) to the phosphonic acid group.

4. **Construct the SMILES**: The backbone HOOC-CH_2-NH-CH_2-P(=O)$(OH)_2$ translates to the following SMILES representation considerations:
   - A carbon chain connecting the amine and the phosphonic acid,
   - A phosphonic acid group,
   - An amino group on the glycine backbone.

5. **Final SMILES representation**:
   - The SMILES string for glyphosate (**N−(phosphonomethyl)glycine**) is:

<Answer>O=C(O)CNCP(=O)(O)O</Answer>

SMILES: O=C(O)CNCP(=O)(O)O

Table 6: An example of the dataset is Glyphosate ($C_3H_8NO_5P$), which consists of a molecule description, reasoning steps (CoT), and the corresponding SMILES representation. The generated SMILES is enclosed within <Answer> tags. The CoT first identifies the core structure of Glyphosate based on the description, then identifies components of this molecule, including glycine, the phosphonomethyl group, and N-alkylation, and identifies functional groups. Finally, it constructs the SMILES representation based on the above decomposition.

of the reasoning path, we construct a messed-up CoT dataset by reshuffling the words within each sample to disrupt the reasoning path. This dataset contains the same words as the corrected CoT data but differs in the reasoning path.

After this construction process, we obtain six control-group datasets: corrected CoT, CoT with minor issues, CoT with element issues, CoT with structural issues, ineffective CoT, and messed-up CoT, each containing $1{,}000$ samples. Since these datasets are randomly selected from the MolCoT-4K training set, we continue to use the MolCoT-4K test and validation sets as the MolCoT-1K test and validation sets.

H Label Distribution of CoT data

The distribution of the CoT label can be seen in Figure 7. The results reveal that the ineffective CoT data rate is the lowest, which means that most of the distilled data is effective. The following rate is correct CoT data, which only 8%, which shows the importance of correcting distilled CoT data. Most of the data (41%) contain minor issues, such as the number of bonds between the carbon chain.

I Label Distribution of Test Data after Supervised Fine-Tuning and Reinforcement Learning

The results are presented in Figure 8. After SFT, the model can generate valid CoT with an accuracy

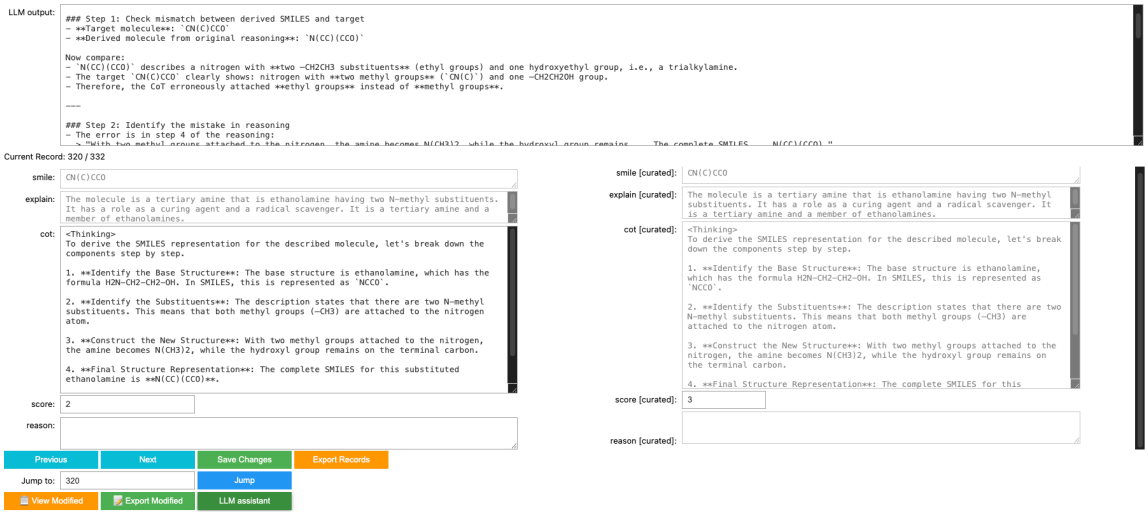


Figure 6: The user interface of the curation tool. The original CoT is presented in the left text box and can be edited. The corrected CoT is presented in the right text box. The corresponding output of assistant LLMs is presented in the top box. These buttons are used to navigate through the data, save, and export records.

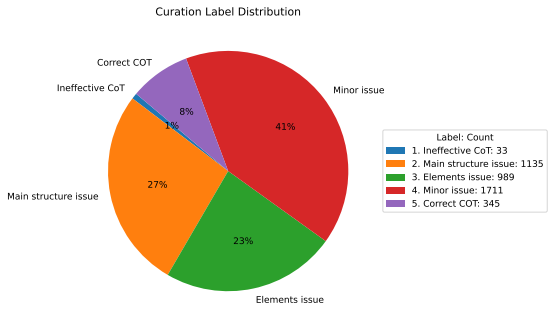


Figure 7: The label distribution of the distilled CoT data.Only 8% is correct CoT data, while 92% of the distilled CoT contains issues, with 1% not even generating valid CoT.

of 99%, and the corrected CoT rate reaches 16%. The mean length of the CoT is $1,099$. After RL, the model can generate valid CoTs with nearly 100% accuracy, and the corrected CoT rate further increases to 18%. The mean length of the CoT is $1,171$. After RL, the quality of the chain-of-thought (CoT) reasoning, in terms of correctness, is also enhanced. Examples of results after RL are presented in Appendix L.

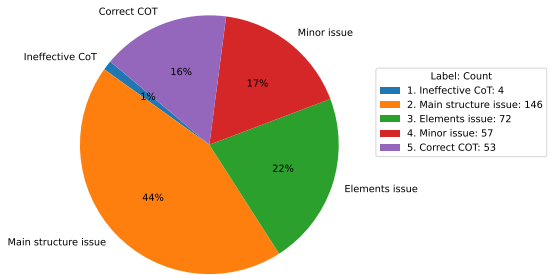
J Supervised Fine-Tuning Implementation Details

We conduct full-parameter supervised fine-tuning of a pretrained causal language model for molecular SMILES generation from natural-language descriptions. Training instances are formatted as chat-style inputs with a system message, a user prompt, and an assistant response. The assistant output is supervised to follow a strict structured format, consisting of a reasoning block enclosed in `<Thinking>` tags and a final SMILES string enclosed in `<Answer>` tags, followed by an explicit EOS token.

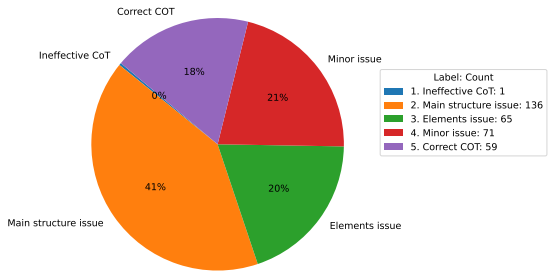
Training and validation data are provided in JSONL format. During preprocessing, answer tags are normalized, and only the last `<Answer>` span is retained as the supervision target; if no answer span is present, the reference SMILES is used instead. Inputs are tokenized using the model-specific chat template and truncated to a maximum length of $2,048$ tokens.

All model parameters are fine-tuned using TRL's SFTTrainer. We enable scaled dot-product attention and train in bfloat16 precision, with model weights automatically sharded across available GPUs. Optimization is performed using AdamW with a learning rate of 5×10^{-5} for 3 epochs. The per-device training batch size is set to 1, with no gradient accumulation. We apply gradient clipping with a maximum norm of 0.3, use a warmup ratio of 0.03, and adopt a constant learning rate schedule. Intermediate checkpoint saving is disabled, and training metrics are logged at every step via Weights & Biases.

Evaluation is performed at the end of each epoch using deterministic decoding without sampling (beam size 1). Generated outputs are post-processed to extract the final `<Answer>` span and evaluated using RDKit for SMILES validity, exact match, and Tanimoto similarity based on Morgan fingerprints (radius 2). Evaluation is decoupled



(a) Label distributions on the test set after SFT.



(b) Label distributions on the test set after SFT and RL.

Figure 8: Test set label distributions. After SFT, the model can generate valid CoT with an accuracy of 99%, with 16% correct CoT. After RL, the model can generate valid CoT with nearly 100% accuracy, with 18% correct CoT. After RL, the quality of the chain-of-thought (CoT) reasoning, in terms of correctness, is also enhanced.

from training and does not affect gradient updates.

K Reinforcement Learning Implementation Details

We further optimize the SMILES generator with reinforcement learning using TRL's GRPOTrainer. Each instance is converted into a chat prompt with a system instruction and a user query; the model is trained to output <Thinking> and <Answer> blocks, and rewards are computed from the extracted last <Answer> span. The policy is initialized from a pretrained causal LM (default: Qwen/Qwen2.5-7B) and loaded with device_map="auto" for multi-GPU execution. Training uses bfloat16 and the AdamW optimizer.

We use the following GRPO hyperparameters (defaults): learning rate 5×10^{-6}, Adam betas $(0.9, 0.99)$, weight decay 0.1, warmup ratio 0.1, cosine LR schedule, per-device batch size 4 with gradient accumulation 1, and gradient clipping with a maximum norm 0.1. For each prompt, we sample num_generations=2 completions with temperature 0.7, using max_prompt_length=$1,024$ and max_completion_length=$1,024$. We save checkpoints every epoch and log to Weights & Biases every step. We also apply a repetition penalty of 1.2 and use loss_type=dr_grpo with truncated-completion masking.

Rewards are RDKit-based. We validate predicted SMILES via RDKit canonicalization and compute chemical equivalence/similarity using Morgan fingerprints (radius 2) with Tanimoto similarity. We employ either (i) a conditional reward scheme combining a strict format reward (2.0) with a stacked EM-gated reward (activated only if both <Thinking> and <Answer> blocks are present and the SMILES is valid), or (ii) a non-conditional additive scheme including strict/soft format rewards, SMILES validity, length shaping, and an EM-based correctness reward (4.0 for exact match; otherwise, 0). After training, we evaluate deterministically (greedy decoding, no sampling) on a held-out test file and report EM, mean Tanimoto similarity, and the SMILES validity rate.

L Examples of Results after Reinforcement Learning

Examples are presented in Table 7. After reinforcement learning (RL), the model is able to generate reasoning steps and produce the corresponding SMILES based on the description: it identifies the core structure and substitutes, determines their positions, assembles the structure, and summarizes before providing the final answer.

M Supervised Fine-Tuning Prompt

The prompt used for Supervised Fine-Tuning is shown in Figure 9. The {Molecule Description}, {molecule generation CoT} and {ground truth SMILES} are replaced by actual data samples.

N Reinforcement Learning Prompt

The prompt used for reinforcement learning is shown in Figure 10. The {Molecule Description} is replaced by actual data samples.

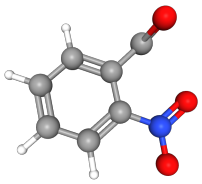
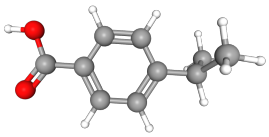
Molecule Structure&Name	Description	Chain of Thought
 2-Nitrobenzaldehyde	The molecule is benzaldehyde substituted at the ortho-position with a nitro group. It is a C-nitro compound and a member of benzaldehydes.	<Thinking> To create the SMILES representation for the described molecule, we'll follow these steps: 1. **Identify the base structure**: The base structure is benzaldehyde, which consists of a benzene ring (C6H5) with an aldehyde group (-CHO) attached. The SMILES for benzaldehyde is c1ccccc1C=O (aromatic) or its Kekulé equivalent C1=CC=CC=C1C=O. 2. **Identify the substituents**: The molecule is substituted at the ortho position with a nitro group (-NO2). The nitro group must be written correctly in SMILES as [N+](=O)[O-]. 3. **Determine the positions**: The ortho position refers to the two carbon atoms adjacent to the carbonyl carbon of the aldehyde group. In benzaldehyde, these are the ortho positions relative to the aldehyde group. 4. **Combine the components**: Attach the nitro group at the ortho position relative to the aldehyde substituent on the benzene ring. Now that we have a clear understanding of the structure, we can convert this into its SMILES notation: </Thinking> <Answer>C1=CC=C(C(=C1)C=O)[N+](=O)[O-]</Answer>
 4-Isopropylbenzoic Acid	The molecule is a cumic acid that consists of benzoic acid substituted by an isopropyl group at position 4. It has a role as a plant metabolite. It is a conjugate acid of a p-cumate.	<Thinking> To derive the SMILES representation for the described molecule, we can break down the components mentioned in the molecular description: 1. **Benzoic Acid**: This is a benzene ring (C6H5) with a carboxylic acid (COOH) group attached. The SMILES for benzoic acid is `c1ccccc1C(=O)O` or Kekulé form `C1=CC=CC=C1C(=O)O`. 2. **Substituted by an Isopropyl Group at Position 4**: An isopropyl group (-CH(CH_3)$_2$) is a branched alkyl group. In SMILES, the isopropyl group can be represented as CC(C). 3. **Combining the Components**: Since the isopropyl group is substituted at the -4position of the benzoic acid, the -4 position refers to the carbon opposite the carboxylic acid on the benzene ring (para position). Thus, the isopropyl group is attached to the carbon para to the -COOH. Now that we have a clear understanding of the structure, we can convert this into its SMILES notation: </Thinking> <Answer>CC(C)C1=CC=C(C=C1)C(=O)O</Answer>

Table 7: Examples of the results. The model generates reasoning steps and finally gives the corresponding SMILES based on the description. After RL, the model identifies the core structure and substitutes, determines their positions, assembles the structure, and summarizes before providing the final answer.

User: You are an expert in info-chemistry, especially in SMILES interpretation and translation. Provide the SMILES representation based on the following molecule description: {Molecule Description}. Analyze it step by step within the <Thinking></Thinking> tags and give only the SMILES within the <Answer></Answer> tags.

Assistant:
<Thinking>
{molecule generation CoT}
</Thinking>
<Answer>
{ground truth SMILES}
</Answer>

Figure 9: Supervised Fine-Tuning Prompt

User: You are an expert in info-chemistry, especially in SMILES interpretation and translation. Provide the SMILES representation based on the following molecule description: {Molecule Description}. Analyze it step by step within the <Thinking></Thinking> tags and give only the SMILES within the <Answer></Answer> tags.

Figure 10: Reinforcement Learning Prompt

O Post-training Details

This section outlines the training approach used to elicit chemical reasoning ability in LLMs. We adopt a two-stage post-training strategy: First, we apply SFT using (m_i, c_i, s_i) in MolCoT-4K as a warm-up stage to equip the model with basic accuracy and the ability to generate CoT responses. Note that the model is supervised by c and s given a description m. We then apply reinforcement learning to further improve the model's reasoning capability.

We apply the widely used GRPO (Shao et al., 2024) reinforcement learning method to further improve the model's reasoning ability after SFT. GRPO discards the value function and estimates the advantage in a group-wise normalization. GRPO typically introduces a KL penalty term to regularize divergence between the online and reference policies, while we eliminate the KL penalty term to allow the policy to explore more aggressively in the chemical space.

Formally, for a pair of question-answer pair $(q, a) \in D$, GRPO method samples a group of G responses $\{o_i\}_{i=1}^{G}$ from the old behavior policy $\pi_{\theta_{\text{old}}}$. A reward model is then used to score SMILES and format correctness, yielding group rewards $\{r_i\}_{i=1}^{G}$. Outcome supervision provides the normalized reward at the end of each output o_i and sets the advantages $\hat{A}_{i,t}$ of all tokens in the output as the normalized reward:

$$\hat{A}_{i,t} = \widetilde{r}_i = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^{G})}{\text{std}(\{r_i\}_{i=1}^{G})}. \tag{1}$$

The resulting GRPO loss without KL penalty is:

$$\begin{aligned} \mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\alpha_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}(\alpha_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\right]\right\}, \end{aligned} \tag{2}$$

where

$$\alpha_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}. \tag{3}$$

ε is a hyper-parameter. Here, we use the molecule description and SMILES pair, (m_i, s_i) in any molecule generation dataset as q and a, respectively.

We follow prior RL applications to math reasoning (Shao et al., 2024; Yu et al., 2026; Liu et al., 2025) and design rewards based on correctness and output format constraints. The correctness reward ensures that the model receives a reward only when the generated SMILES exactly matches the ground-truth SMILES[5]. The format reward ensures that the model receives a reward only when the generated CoT contains both the `<Think>` and `<Answer>` tags.

However, for molecule generation, correctness and formatting alone are insufficient due to the complexity of chemical syntax and combinations. We therefore design a molecule-specific ensemble reward strategy, where we additionally introduce a SMILES validity reward that ensures the generated output corresponds to a chemically valid molecule, verified by the RDKit tool.

[5]Our preliminary experiments show minimal gains when using similarity-based rewards, e.g., those from the RDKit tool (Landrum et al., 2025). Thus, we chose the simplest binary reward version.