Contents

Supplementary Section 1: Data Analysis of ChemToolDataset 2
Supplementary Section 2: Multi-Agent System Performance on ChemCoTBench 3

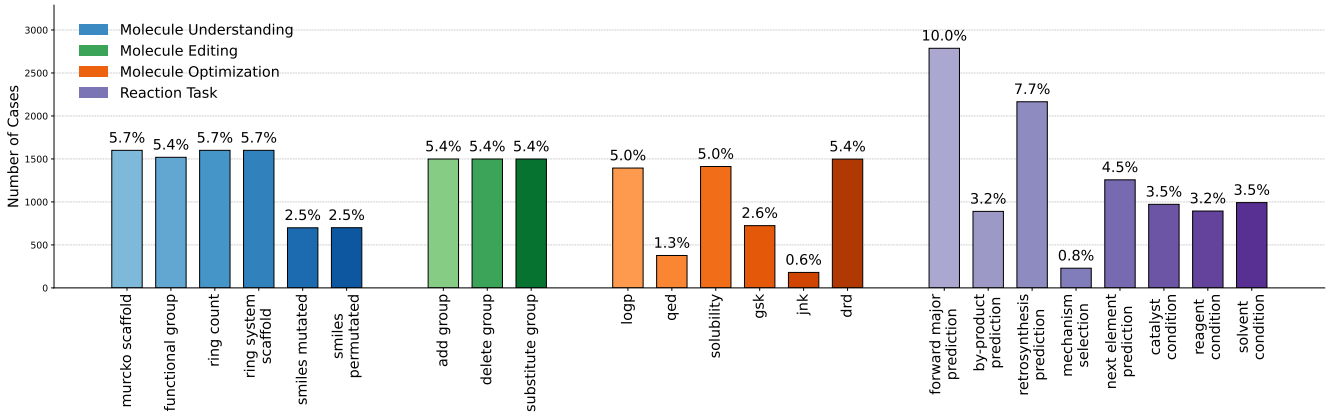


Figure 1. Composition of ChemToolDataset across four major task families and their subtasks. The dataset contains 27,986 samples, with reaction tasks forming the largest portion, followed by molecule understanding, molecule optimization, and molecule editing.

Supplementary Section 1: Data Analysis of ChemToolDataset

ChemToolDataset is constructed as a tool-integrated extension of ChemCoTDataset[1]. While preserving the original four-category taxonomy, including molecule understanding, molecule editing, molecule optimization, and reaction reasoning, ChemToolDataset further converts static chain-of-thought annotations into agent-interacted reasoning trajectories. This design is intended to better reflect practical chemical problem solving, where a model must not only reason over molecular and reaction information in text, but also invoke external computational tools to verify structures, retrieve reaction evidence, calculate molecular properties, and refine intermediate hypotheses.

Specifically, ChemToolDataset improves the original ChemCoTDataset through a Hypothesis-Action-Refinement construction pipeline. In the hypothesis stage, the teacher model identifies missing chemical evidence or uncertain reasoning steps in the original trajectory and determines where tool calls should be inserted. In the action stage, the Chemical Agent Sandbox executes the corresponding chemical tools, such as RDKit-based structure parsing, property prediction, and reaction-related retrieval. In the refinement stage, the tool outputs are reintegrated into the reasoning process, producing fluent and logically grounded agent trajectories. As a result, ChemToolDataset provides supervision not only for chemical reasoning[2], but also for when and how to use tools during scientific problem solving[3].

Dataset composition. As shown in Fig. 1, ChemToolDataset contains 27,986 samples across four major task families. Reaction tasks constitute the largest portion, with 10,186 samples, accounting for 36.4% of the dataset. This emphasis reflects the central role of tool use in reaction reasoning, where models often need to combine substrate analysis, reaction rule recognition, condition inference, and mechanistic validation. Molecule understanding contains 7,718 samples, corresponding to 27.6%, and provides the structural foundation for downstream chemical reasoning. Molecule optimization contributes 5,585 samples, or 20.0%, covering both physicochemical property improvement and target-oriented molecular design. Molecule editing contains 4,497 samples, or 16.1%, focusing on controllable molecular modifications through functional-group addition, deletion, and substitution.

At the subtask level, ChemToolDataset maintains broad coverage of fundamental and application-oriented chemical capabilities. Molecule understanding[4] includes functional group detection, ring counting, Murcko scaffold recognition, ring system scaffold recognition, and SMILES robustness tasks based on mutation and permutation. Molecule editing balances three elementary operations, namely adding, deleting, and substituting functional groups. Molecule optimization[5,6] covers LogP, QED, solubility, and target-related objectives, including GSK3, JNK3, and DRD2. Reaction tasks span forward major-product prediction[7], by-product prediction[8], retrosynthesis[9], mechanism selection[10], next elementary-step prediction, and reaction condition recommendation[11] for catalysts, reagents, and solvents.

Overall, the distribution of ChemToolDataset reflects a deliberate balance between foundational molecular understanding and higher-level chemical decision making. The relatively large proportion of reaction and optimization samples exposes models to tasks where external tools are especially valuable, while the understanding and editing subsets ensure that the model retains fine-grained control over molecular structures. This composition makes ChemToolDataset suitable for training and evaluating chemistry agents that must coordinate textual reasoning, molecular manipulation, property analysis, and tool-grounded verification.

Supplementary Section 2: Multi-Agent System Performance on ChemCoTBench

Multi-agent systems (MAS) provide a system-centric approach for constructing chemistry-oriented AI assistants[12–14]. Rather than asking a single language model to directly solve a complete scientific problem, an MAS decomposes the workflow into multiple interacting agents, such as a planner, a chemical tool executor, a verifier, and a final response synthesizer. Formally, a multi-agent chemical reasoning system can be written as

$$\mathcal{M} = \langle \mathcal{A}, \mathcal{T}, \mathcal{E}, \mathcal{C}, \Pi \rangle, \tag{1}$$

where $\mathcal{A} = \{A_1, \ldots, A_n\}$ denotes a set of LLM-driven agents, $\mathcal{T}$ is the chemical tool set, $\mathcal{E}$ is the execution environment, $\mathcal{C}$ is the communication protocol among agents and tools, and $\Pi = \{\pi_1, \ldots, \pi_n\}$ denotes the agent policies. Given a chemical query x, the system generates an interactive trajectory

$$\tau = \{(s_t, a_t, o_t)\}_{t=1}^{T}, \quad a_t \sim \pi_{i_t}(\cdot \mid s_t), \quad o_t = \mathcal{E}(a_t), \tag{2}$$

where s_t is the intermediate reasoning state or message context, a_t is the action selected by agent A_{i_t}, and o_t is the observation returned by an external tool, database, or another agent. The final answer is then produced by an aggregation function $y = f(\tau)$. This formulation makes MAS flexible and easy to extend, but its final performance depends on whether every intermediate agent decision, tool invocation, and message transfer remains correct.

We evaluate this paradigm on ChemCoTBench using ChemCrow[15] and SciToolAgent[16] as representative multi-agent systems. The comparison covers four categories of chemistry tasks: molecular structural understanding, molecule editing, molecule optimization, and reaction prediction. These tasks jointly examine whether an agent can perceive molecular structures, execute chemically valid transformations, improve molecular properties, and reason over synthesis-related outcomes. As shown in Table 1, ChemCrow and SciToolAgent exhibit highly uneven behavior on molecule understanding and editing. ChemCrow obtains moderate errors on functional-group and ring counting (0.33 and 0.55), but its performance drops on SMILES equivalence (59) and molecule editing, with success rates of only 42.8%, 45.4%, and 49.0% for add, delete, and substitute operations. SciToolAgent shows a different failure pattern: it achieves a high SMILES equivalence score (89) and competitive substitution performance (80%), but produces much larger counting errors (0.91 for functional groups and 1.75 for rings) and completely fails on the add operation (0%). In contrast, ChemCRAFT achieves lower counting errors (0.06 and 0.15), stronger SMILES equivalence (97), and more reliable editing performance, especially on add (90.0%) and delete (100.0%).

Table 2 further demonstrates the instability of multi-agent systems in molecule optimization. ChemCrow is able to improve solubility substantially ($\Delta = 2.54$, SR=60%), but this gain does not generalize to other objectives: it degrades LogP ($\Delta = -1.74$), QED ($\Delta = -0.06$), DRD2 ($\Delta = -0.06$), JNK3 ($\Delta = -0.13$), and GSK3-β ($\Delta = -0.09$). SciToolAgent records zero improvement and zero success rate across all six optimization objectives, indicating that its interaction pipeline fails to convert tool access into effective property-guided molecular modification. By comparison, ChemCRAFT provides more consistent improvements on physicochemical properties, including LogP ($\Delta = 0.60$, SR=78%), solubility ($\Delta = 0.93$, SR=94%), and QED ($\Delta = 0.15$, SR=81.8%). Although protein activity-related objectives remain more challenging, ChemCRAFT still maintains non-trivial success rates, showing that the unified agent policy is more robust than manually coordinated multi-agent workflows.

The reaction prediction results in Table 3 show a similar pattern. ChemCrow performs weakly across major-product prediction, by-product prediction, retrosynthesis, and condition prediction, with very low Top-1 accuracy and fingerprint similarity. SciToolAgent achieves strong performance on forward major-product prediction (Top-1=0.69, FTS=0.66), but this advantage collapses on other reaction subtasks: it obtains 0.0 Top-1 on by-product prediction, 0.03 on retrosynthesis, and 0.0 on condition prediction. This indicates that multi-agent systems can be effective when a subtask matches a specific tool chain, but they do not necessarily provide stable generalization across different chemical reasoning modes. ChemCRAFT achieves consistently stronger results across all reaction subtasks, including major-product prediction (Top-1=0.83, FTS=0.85), by-product prediction (Top-1=0.56, FTS=0.39), retrosynthesis (Top-1=0.41, FTS=0.45), and condition prediction (Top-1=0.90, FTS=0.91).

We attribute this instability to error accumulation during LLM-based multi-agent interaction. Suppose a multi-agent workflow requires T reasoning, communication, or tool-use steps, and the error probability at step t is ε_t. Even under a simplified independence assumption, the probability that the entire trajectory remains correct is

$$P(\text{success}) \approx \prod_{t=1}^{T} (1 - \varepsilon_t), \tag{3}$$

and the trajectory-level failure probability becomes

$$P(\text{failure}) = 1 - \prod_{t=1}^{T} (1 - \varepsilon_t) \approx \sum_{t=1}^{T} \varepsilon_t, \tag{4}$$

when each ε_t is small. In practice, the situation is more severe because errors are not independent. An early mistake in functional-group recognition, invalid SMILES parsing, irrelevant reaction-template retrieval, or ambiguous natural-language communication can bias subsequent agents and contaminate later tool calls. Moreover, multi-agent systems introduce additional overhead from role hand-offs, repeated planning, format conversion, and final answer aggregation. The results in Tables 1, 2, and 3 therefore suggest that simply increasing the number of agents or tools does not guarantee reliable chemical intelligence. A more stable solution requires optimizing the tool-use policy itself, so that planning, tool invocation, observation interpretation, and final reasoning are aligned within a compact and trainable agentic workflow.

Table 1. The performance analysis of multi-agent system (ChemCrow, SciToolAgent) on the molecule understanding and molecule editing tasks.

Models	Func-Group		Scaffold		SMILES	Molecule-Edit		
	FG↓	Ring↓	Murcko↑	Ring-sys↑	Eq.↑	Add↑	Delete↑	Sub↑
Qwen3-235B-A22B	0.42	1.00	0.34	82.5	75	40.0	75.0	66.7
Qwen2.5-32B-Instruct	0.36	0.65	0.12	53.3	62	50.0	50.0	48.3
ChemCrow	0.33	0.55	0.31	0.52	59	42.8	45.4	49.0
SciToolAgent	0.91	1.75	0.14	0.273	89	0	50	80
ChemCraft	**0.06**	**0.15**	0.26	**100**	**97**	90.0	**100.0**	73.0

Table 2. Baseline Performance on Molecule Optimization. The optimized targets are categorized into physicochemical properties (QED, LogP, solubility) and protein activity-related properties (JNK3, DRD2, GSK-3β), with the latter posing greater challenges to the model's chemical knowledge and reasoning capabilities. Δ is the mean property improvement, where a negative Δ indicates that most optimizations are property degradations. SR% is the success rate that brings property increase.

Models	LogP		Solubility		QED		DRD2		JNK3		GSK3-β	
	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%	Δ	SR%
Qwen3-235B-A22B	0.03	21	0.18	45	0.07	34	0.04	26	-0.01	18	0.02	25
Qwen2.5-32B-Instruct	0.15	44	0.49	65	0.09	54	0.05	32	-0.02	19	0.01	31
ChemCrow	-1.74	9	2.54	60	-0.06	9	-0.06	17	-0.13	6	-0.09	6
SciToolAgent	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
ChemCraft	0.60	78	0.93	94	0.15	81.8	0.03	48.4	-0.03	20.6	0.01	38.1

Table 3. The chemical reaction task contains forward prediction ($\text{Fwd}_{\text{major}}$: major-product prediction, and Fwd_{by}: by-product prediction), resynthesis prediction (Retro), and reaction condition prediction (Condition). FTS: molecule fingerprint similarity with reference.

Models	Fwd $_{\text{major}}$		Fwd $_{\text{by}}$		Retro		Condition	
	Top-1	FTS↑	Top-1	FTS↑	Top-1	FTS↑	Top-1	FTS↑
Qwen3-235B-A22B	0.04	0.57	0.0	0.06	0.0	0.30	0.07	0.14
Qwen2.5-32B-Instruct	0.01	0.43	0.0	0.12	0.0	0.29	0.02	0.10
ChemCrow	0.01	0.05	0.04	0.04	0.0	0.02	0.06	0.07
SciToolAgent	0.69	0.66	0.0	0.03	0.03	0.05	0.0	0.01
ChemCraft	0.83	0.85	0.56	0.39	0.41	0.45	0.90	0.91

References

1. Li, H. *et al.* Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations (2026). 2505.21318.
2. Bagal, V., Aggarwal, R., Vinod, P. & Priyakumar, U. D. Molgpt: molecular generation using a transformer-decoder model. *J. chemical information modeling* **62**, 2064–2076 (2021).
3. Boiko, D. A., MacKnight, R., Kline, B. & Gomes, G. Autonomous chemical research with large language models. *Nature* **624**, 570–578 (2023).
4. Zeng, Z., Yao, Y., Liu, Z. & Sun, M. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nat. communications* **13**, 862 (2022).
5. Chen, X., Wu, R., Lan, Y., Ma, T. & Liu, Y. Molevolve: Llm-guided evolutionary search for interpretable molecular optimization (2026). 2603.24382.
6. Liu, H.-P., Phoa, F. K. H. & Dutta, S. Molecule discovery and optimization via evolutionary swarm intelligence. *Sci. Reports* **14**, 24510 (2024).
7. Coley, C. W., Barzilay, R., Jaakkola, T. S., Green, W. H. & Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS central science* **3**, 434–443 (2017).
8. Yan, Y. *et al.* Rpbp: deep retrosynthesis reaction prediction based on byproducts. *J. Chem. Inf. Model.* **63**, 5956–5970 (2023).
9. Wang, Y. *et al.* Retrosynthesis prediction with an interpretable deep-learning framework based on molecular assembly tasks. *Nat. Commun.* **14**, 6155 (2023).
10. Kayala, M. A. & Baldi, P. Reactionpredictor: prediction of complex chemical reactions at the mechanistic level using machine learning. *J. chemical information modeling* **52**, 2526–2540 (2012).
11. Zhang, Y. *et al.* Text-augmented multimodal llms for chemical reaction condition recommendation. *arXiv preprint arXiv:2407.15141* (2024).
12. Lyu, Y. *et al.* Evoscientist: Towards multi-agent evolving ai scientists for end-to-end scientific discovery. *arXiv preprint arXiv:2603.08127* (2026).
13. Song, T. *et al.* A multiagent-driven robotic ai chemist enabling autonomous chemical research on demand. *J. Am. Chem. Soc.* **147**, 12534–12545 (2025).
14. Zeng, T., Badrinarayanan, S., Ock, J., Lai, C.-K. & Barati Farimani, A. Llm-guided chemical process optimization with a multi-agent approach. *Mach. Learn. Sci. Technol.* **6**, 045067 (2025).
15. Bran, A. M. *et al.* Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376* (2023).
16. Ding, K. *et al.* Scitoolagent: a knowledge-graph-driven scientific agent for multitool integration. *Nat. Comput. Sci.* **5**, 962–972 (2025).