

ChemAmp: Amplified Chemistry Tools via Composable Agents

Zhucong Li^{1,3*}, Powei Chang^{2*}, Jin Xiao², Zhijian Zhou¹,

Qianyu He³, Jiaqing Liang², Fenglei Cao⁴, Xu Yinghui¹, Yuan Qi^{1,4,5†}

¹Artificial Intelligence Innovation and Incubation Institute, Fudan University,

²School of Data Science, Fudan University,

³College of Computer Science and Artificial Intelligence, Fudan University,

⁴Shanghai Academy of Artificial Intelligence for Science,

⁵Department of Information and Intelligence Development, Zhongshan Hospital, Fudan University

{zcli22, bwzhang24, jinxiao23}@m.fudan.edu.cn, {liangjiaqing, xuyinghui, qi yuan}@fudan.edu.cn

Abstract

Although LLM-based agents are proven to master tool orchestration in scientific fields, particularly chemistry, their single-task performance remains limited by underlying tool constraints. To this end, we propose tool amplification, a novel paradigm that enhances the collective capabilities of specialized tools through optimized, dynamic coordination within individual tasks. Instantiating this paradigm, we introduce ChemAmp, a computationally lightweight framework that dynamically treats chemistry tools (e.g., UniMol2, Chemformer) as composable building-block agents. It constructs task-specialized super-agents that transcend atomic tool constraints with limited data (≤ 10 samples). Our evaluations across four core chemistry tasks molecular design, molecule captioning, reaction prediction, and property prediction demonstrate that ChemAmp outperforms chemistry-specialized models, generalist LLMs, and agent systems with tool orchestration. Critically, this bottom-up construction strategy enables 94% inference token cost reductions versus vanilla multi-agent systems. Our code and dataset are publicly available at <https://github.com/Chang-pw/ChemAmp>.

1 Introduction

Large Language Model (LLM)-based agents (Touvron et al., 2023; Achiam et al., 2023; Guo, 2025) have emerged as powerful tools for automating complex scientific workflows, particularly in chemistry, where they orchestrate multi-step processes involving specialized computational tools (Bran et al., 2023; Ouyang et al., 2024; Boiko et al., 2023; Han et al., 2024; Shi et al., 2023; Yu et al., 2025; Tang et al., 2025).

As shown in Fig 1 (Left), while effective for task orchestration, current works primarily sequence pre-defined tool usage. For example, when an

LLM-based agent handles three potential pipeline tasks such as molecular design (Noutahi et al., 2023), reaction prediction (Shi et al., 2023), and property prediction (Srinivas and Runkana, 2024), it sequentially selects the predefined chemistry tools for each task to obtain computational results. However, a critical limitation remains unaddressed: the performance of these agents on individual, complex tasks is fundamentally constrained by the inherent capabilities and scope of the underlying tools they invoke. As a result, errors can propagate through the reasoning chain. This bottleneck restricts performance gains and often leads to inefficient, redundant tool calls that inflate computational costs.

In this work, we confront a more fundamental limitation: even state-of-the-art chemistry tools (e.g., UniMol2, Chemformer, ChemDFM) (Ji et al., 2024b; Irwin et al.; Zhao et al., 2024) remain constrained by their atomic capabilities when operating in isolation. As illustrated in Fig. 1 (Right), we posit a fundamental shift from tool orchestration to tool amplification where tools transcend native capabilities through hierarchical coordination within atomic tasks. Unlike traditional tool orchestration (which schedules tools across different tasks), amplification focuses on making tools work better together within a single task. By dynamically combining tools into collaborative teams, they can achieve capabilities beyond what any tool can do alone.

This amplification paradigm introduces two core challenges: (1) Structural Search Complexity: Optimal agent-tool compositions vary significantly across chemistry tasks due to domain-specific constraints. (2) Efficiency-Capability Tradeoff: Manual composition is infeasible, while exhaustive search incurs prohibitive computational costs.

To address these challenges, we introduce **ChemAmp (Chemistry Tool Amplification)**, a framework realizing tool amplification through

* Equal contribution.

† Corresponding author.

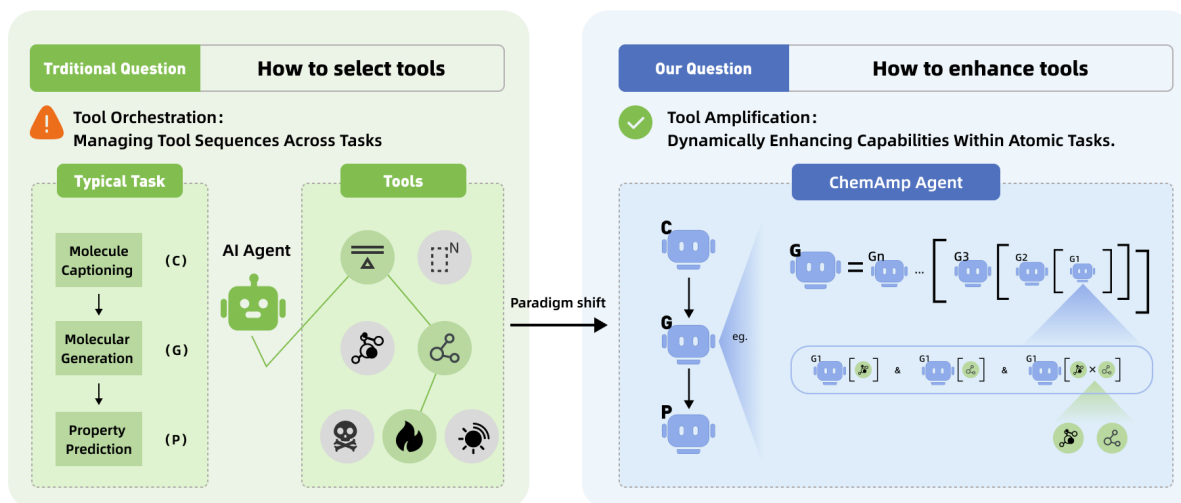


Figure 1: The research framework. This work introduces a paradigm shift—from tool orchestration (managing tool sequences across tasks) to tool amplification (dynamically enhancing capabilities within atomic tasks).

composable agents. Departing from naive stacking, ChemAmp employs a **bi-phase encapsulation engine** constructing agent hierarchies from atomic tools via bottom-up iterative composition. Specifically, ChemAmp operates through two synergistic stages: (1) **Atomic-to-Composite Amplification**: Atomic tools undergo iterative encapsulation into sub-agents, evolving optimal combinations through adaptive scoring and automated feedback. (2) **Cross-Composite Synergy**: Sub-agents are further encapsulated into composite networks through the same mechanism, amplifying capabilities via emergent hierarchical coordination.

We deploy ChemAmp across four foundational chemistry domains: molecular design, molecule captioning, property prediction, and reaction prediction. For each task, ChemAmp discovers optimal agent compositions, dynamically constructing task-specialized super-agents that transcend atomic tool constraints with limited data (≤ 10 samples). Experimental results demonstrate the consistent superiority of ChemAmp over three critical baselines: chemistry-specialized (Ji et al., 2024a; Zhao et al., 2024), generalist LLMs (OpenAI and et al., 2024; Guo, 2025), and agent systems with tool orchestration (Bran et al., 2023; Yu et al., 2025). Notably, ChemAmp delivers significant gains while requiring only 6% inference token costs versus basic multi-agent systems, validating both the efficacy and efficiency of tool amplification. This paradigm shift enables ChemAmp to automatically construct task-specialized super-agents that amplify tool capabilities.

To summarize, our contributions are three-fold:

- We propose tool amplification, a novel paradigm that enhances the collective capabilities of specialized tools through optimized, dynamic coordination within individual tasks.
- We develop ChemAmp, a lightweight framework realizing amplification via bi-phase iterative encapsulation of atomic tools into agent composite tools.
- We demonstrate state-of-the-art results across molecular design, captioning, reaction prediction, property prediction and 94% inference token cost reductions versus vanilla multi-agent systems.

2 Related Work

2.1 Orchestration Paradigm

The foundation for computational chemistry agents builds on the ReAct framework (Yao et al., 2023), which integrates reasoning with tool execution. This paradigm has evolved into sophisticated tool orchestration systems: (1) Workflow-focused: ChemCrow (Bran et al., 2023) and Co-scientist (Boiko et al., 2023) sequence tools like RDKit/LabX across multi-step tasks. (2) Domain-optimized: ChemAgent (Tang et al., 2025) and SciToolAgent (Chen et al., 2025) improves LLM performance in complex chemical reasoning tasks by introducing a dynamic, self-evolving memory library that supports task decomposition and solution generation. ChemToolAgent (Yu et al., 2025)

Example For The Task Molecular Generation (G)

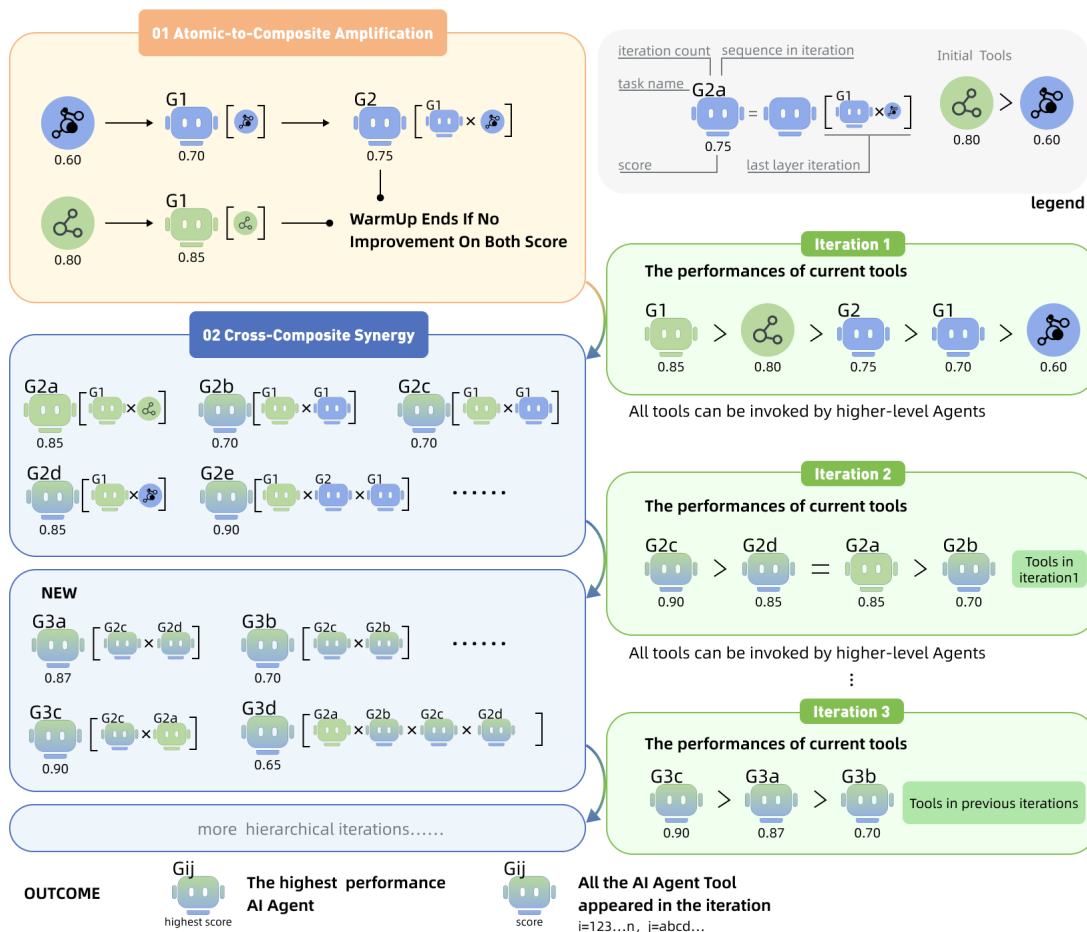


Figure 2: ChemAmp’s two-stage amplification process: (1) Atomic-to-Composite Amplification: atomic tools are encapsulated into agents; (2) Cross-Composite Synergy: agents are merged into hierarchical networks.

supports a large tool set and performs dynamic tool selection in a broad task suite. (3) Hybrid optimization: Recent systems like AgentPrune (Zhang et al., 2024a), GPTSwarm (Zhuge et al., 2024), Aflow (Zhang et al., 2024c), and MaAS (Zhang et al., 2025) automate workflow refinement after-initial manual design. As a generic optimization of manually orchestrated vanilla multi-agent systems, such frameworks deliberately avoid atomic tool-level performance enhancements. Consequently, our experiments compare against multiple instantiations of vanilla multi-agent systems rather than these derivative approaches.

2.2 Paradigm Shift Imperative

Despite these advances, orchestration faces fundamental constraints: (1) Capability Ceiling: Exact accuracy plateaus at 35% (e.g., molecule captioning) (2) Coordination Scope: They remain inefficient in utilizing existing computational chemistry

tools (Shi et al., 2023), and struggle to navigate the combinatorial and hierarchical relationships between these tools.

This impasse necessitates a paradigm shift: from scheduling tools across tasks (orchestration) to amplifying capabilities within tasks through dynamic agent composition. The amplification paradigm introduces two core challenges: (1) Structural Search Complexity: Optimal agent-tool compositions vary significantly across chemistry tasks due to domain-specific constraints. (2) Efficiency-Capability Tradeoff: Manual composition is infeasible, while exhaustive search incurs prohibitive computational costs.

3 ChemAmp Framework

Our study proposes **ChemAmp**, a framework that realizes the tool amplification paradigm through hierarchical composition of composable agents.

dataset for each task in ChemLLMBench contains only 100 instances, we adopt a similar approach to that in (Guo et al., 2023) to select the validation set. For the Text-Based Molecule Design and Molecule Captioning tasks, we randomly sample 100 instances from the ChEBI-20-MM (Liu et al., 2025) dataset, excluding the corresponding test set, as the sample set for validation. For the Reaction Prediction task, we randomly sample 100 instances from the USPTO-MIT (Jin et al., 2017) dataset, excluding the corresponding test set, as the validation set. For the Molecular Property Prediction task, we randomly sample 50 instances for each dataset from the BBBP, HIV, BACE, Tox21, and ClinTox (Wu et al., 2018) datasets, excluding the corresponding test sets, as the validation set. The details of our dataset are shown in supplementary material.

Models We categorize current models into two primary groups: chemistry specialized models and generalist LLMs. chemistry specialized models refer to non-LLM models designed for specific tasks, while generalist LLMs are LLMs primarily designed for general-purpose use across a wide range of tasks. And the agent in the LLM-based model leverage GPT-4o (OpenAI and et al., 2024) as the core agent with ReAct (Yao et al., 2023) framework, augmented with additional tools tailored to the specific task. To ensure fair comparisons, we use the same test set for evaluating different models on each task.

Tools For each task, an initial tool set is provided. The large language model autonomously selects the most suitable tool from the tool set as the base tool to be enhanced. In the following experiments, we report only the best-performing results.

4.2 Results

4.2.1 Text-based Molecule Design

In the text-based molecule design task, LLMs predict a molecule’s SMILES (Simplified Molecular Input Line Entry System) representation based on a given description, testing their ability to interpret and translate chemical language into valid molecular structures (Zhao et al., 2024).

Our study employs two sets of metrics to evaluate the performance of the task. The first set of metrics measures the text-based similarity between the predicted SMILES and the gold standard SMILES, including exact match, BLEU, and Levenshtein

distance (Haldar and Mukhopadhyay, 2011). The second set of metrics assesses the chemical similarity between the predicted molecules and the reference molecules, encompassing the validity of the predicted SMILES and the FTS (Fingerprint Tanimoto Similarity) (Tanimoto, 1958), calculated based on MACCS, RDK, and Morgan (Morgan, 1965).

Model	Exact BLEU	Dis	Validity	MACCS	RDK	Morgan	FTS
<i>Chemistry specialized models</i>							
MolXPT	0.22	-	0.98	0.86	0.76	0.67	0.76
Text+Chem T5	0.32	0.85	16.87	0.94	0.90	0.82	0.75
Mol-Instruction	0.02	0.35	41.40	1.00	0.41	0.23	0.15
ChemDFM-13B	0.32	0.85	<u>11.58</u>	0.94	0.81	0.73	0.67
<i>Generalist LLMs</i>							
GPT-4o	0.01	0.57	52.85	0.91	0.71	0.54	0.38
Deepseek-R1	0.02	0.56	92.29	0.57	0.48	0.38	0.31
Llama3-70b	0.03	0.57	46.63	0.78	0.57	0.40	0.30
<i>Agent with Tool Orchestration</i>							
ChemCrow	0.34	0.77	45.4	0.68	0.84	0.80	0.76
ChemToolAgent	0.28	-	-	0.90	-	-	0.77
<i>Agent with Tool Amplification</i>							
Ours (Stage 1)	<u>0.34</u>	<u>0.87</u>	12.63	0.94	0.85	0.80	0.74
Ours (Stage 1+2)	0.38	0.93	8.68	0.96	0.92	0.87	0.80

Table 1: Benchmark results in text-based molecule design task. All methods are evaluated on 0-shot. Our approach utilizes the chemistry-specialized models listed in this table as tools. ChemCrow and ChemToolAgent additionally employ all tools mentioned in their papers, amounting to 18 and 29 tools respectively.

From the results in Tab. 1, our method consistently outperforms all baselines. It achieves high molecular validity while satisfying target specifications. In contrast, LLM-based models exhibit limited performance, particularly in Exact and BLEU scores, underscoring the difficulty of accurate molecular generation in a 0-shot setting.

4.2.2 Molecule Captioning

To evaluate the model’s capacity to translate complex chemical representations into natural language, we adopt the Molecule Captioning task (Guo et al., 2023), which requires generating concise descriptions from input SMILES strings. To assess the model’s performance on this task, we employ traditional NLP evaluation metrics, such as BLEU and ROUGE, to measure the similarity between the molecule descriptions generated by the model and the reference descriptions in the test set.

As shown in Tab. 2, ChemAmp framework, achieves state-of-the-art performance across all metrics. Chemistry specific models maintain a clear advantage in molecule captioning, while most general-purpose LLMs—except large-scale ones like GPT-4o and DeepSeek-R1—exhibit results.

Model	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
<i>Chemistry specialized models</i>					
Text+Chem T5	0.63	0.54	0.68	0.54	0.62
MolXPT	0.59	0.50	0.66	0.51	0.60
InstructMol	0.48	0.37	0.57	0.39	0.50
Mol-Instruction	0.25	0.17	0.33	0.29	0.27
ChemDFM-13b	0.32	0.27	0.49	0.37	0.48
<i>Generalist LLMs</i>					
GPT-4o	0.26	0.17	0.10	0.00	0.30
Deepseek-R1	0.40	0.25	0.10	0.02	0.21
Llama3-70b	0.11	0.07	0.06	0.00	0.12
<i>Agent with Tool Orchestration</i>					
ChemCrow	0.25	0.20	0.18	0.11	0.23
ChemToolAgent	0.63	0.54	0.45	0.31	0.55
<i>Agent with Tool Amplification</i>					
Ours (Stage 1)	0.64	0.56	0.45	0.29	0.55
Ours (Stage 1+2)	0.73	0.69	0.70	0.58	0.76

Table 2: Benchmark results in molecule captioning task. All methods are evaluated on 0-shot. Our approach utilizes the chemistry-specialized models listed in this table as tools. ChemCrow and ChemToolAgent additionally employ all tools mentioned in their papers, amounting to 18 and 29 tools respectively.

4.2.3 Molecular Property Prediction

Molecular property prediction (Guo et al., 2021; Wang et al., 2021) is a core task in computational chemistry, with broad applications in drug discovery and materials science. This task involves predicting chemical or physical properties from molecular structures. We evaluate performance on five benchmarks from MoleculeNet (Wu et al., 2018): BACE, BBBP, HIV, ClinTox, and Tox21. Considering that our method is based on large language models, we adopt accuracy as the primary evaluation metric.

Model	BACE	BBBP	ClinTox	HIV	Tox21	Avg
<i>Chemistry specialized models</i>						
Uni-Mol-v2	0.75	0.58	0.51	0.96	0.92	0.74
ChemDFM-13B	0.66	0.57	0.49	0.94	0.83	0.70
<i>Generalist LLMs</i>						
GPT-4o	0.38	0.56	0.51	0.59	0.37	0.48
Deepseek-R1	0.62	0.61	0.48	0.51	0.75	0.60
Llama3-70B	0.55	0.59	0.48	0.20	0.59	0.48
<i>Agent with Tool Orchestration</i>						
ChemCrow	0.61	0.60	0.50	0.60	0.77	0.62
ChemToolAgent	-	0.90	0.82	0.94	-	-
<i>Agent with Tool Amplification</i>						
Ours (Stage 1)	0.75	0.59	0.49	0.92	0.94	0.74
Ours (Stage 1+2)	0.79	0.68	0.67	0.96	0.96	0.81

Table 3: Benchmark results in molecular property prediction task. All methods are evaluated on 0-shot. Our approach utilizes the chemistry-specialized models listed in this table as tools. ChemCrow and ChemToolAgent additionally employ all tools mentioned in their papers, amounting to 18 and 29 tools respectively.

As shown in Tab. 3, ChemToolAgent achieved the best performance on the BBBP and Clintox datasets, which benefits from the capabilities of the tools it employs. Conversely, our method achieved state-of-the-art results on the other three datasets. Utilizing only the two open-source models, Uni-Mol-v2 and ChemDFM-13B, our approach demonstrates evident improvements over both models individually.

4.2.4 Reaction Prediction

Reaction prediction is a fundamental task in chemistry, essential for drug discovery, materials science, and the design of novel synthetic routes. Given a set of reactants, the objective is to predict the most probable reaction products (Guo et al., 2024; Schwaller et al., 2019). Consistent with the text-based molecule design task, we employ the same evaluation metrics to assess model performance.

Model	Exact BLEU	Dis	Validity	MACCS	RDK	Morgan	FTS
<i>Chemistry specialized models</i>							
Chemformer	0.91	96.1	1.26	1.00	0.97	0.97	0.97
Text+ChemT5	0.83	96.0	7.42	0.98	0.96	0.96	0.94
InstructMol	0.54	96.7	10.85	1.00	0.88	0.78	0.80
Mol-Instruction	0.05	65.4	27.26	1.00	0.51	0.31	0.26
ChemDFM-13B	0.39	80.6	10.38	0.96	0.77	0.69	0.70
<i>Generalist LLMs</i>							
GPT-4o	0.01	65.8	27.24	0.81	0.54	0.39	0.33
Deepseek-R1	0.10	76.2	16.04	0.75	0.60	0.53	0.48
Llama3-70b	0.00	55.2	282.46	0.85	0.48	0.35	0.31
<i>Agent with Tool Orchestration</i>							
ChemCrow	0.70	90.2	10.11	0.90	0.95	0.93	0.90
ChemToolAgent	0.78	-	-	0.98	-	-	0.98
<i>Agent with Tool Amplification</i>							
Ours (Stage 1)	0.87	97.1	1.00	1.00	0.97	0.97	0.95
Ours (Stage 1+2)	0.91	98.4	0.97	1.00	0.98	0.98	0.96

Table 4: Benchmark results in reaction prediction task. All methods are evaluated on 0-shot. Our approach utilizes the chemistry-specialized models listed in this table as tools. ChemCrow and ChemToolAgent additionally employ all tools mentioned in their papers, amounting to 18 and 29 tools respectively.

As shown in Tab. 4, Chemformer demonstrates strong performance on this task, achieving a product prediction accuracy of 0.91 and outperforming other task-specific models across all metrics. In contrast, LLMs face notable challenges; for instance, DeepSeek-R1 achieves only 0.10 accuracy despite its advanced reasoning capabilities, and ChemDFM performs poorly under zero-shot settings. ChemAmp achieves competitive results across all metrics, surpassing Chemformer in all but Exact score (with only a marginal 0.01 difference).

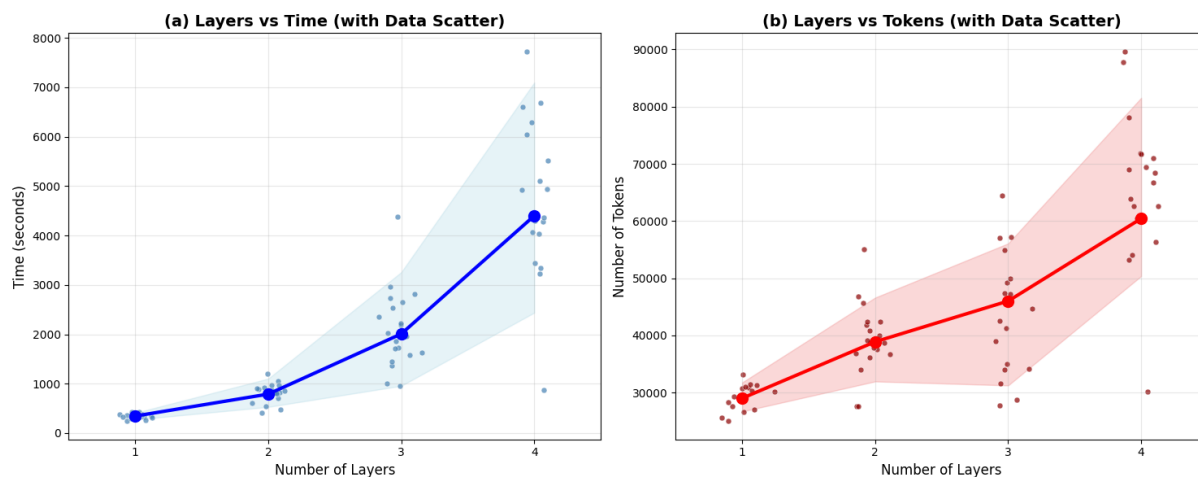


Figure 3: Validation cost (time, tokens) over all tasks with increasing layer depth.

5 Analysis

5.1 Generalization: Few-Shot&Tool-Constrained

To evaluate the influence of tool diversity and validation data size, we conduct experiments on the text-based molecule design task. As shown in Tab. 5, increasing the number of tools from 2 to 4 under a fixed validation set leads to minimal variation in BLEU-2 scores (0.86 for Tool Number = 2 and 4; 0.85 for Tool Number = 3), suggesting diminishing returns from tool addition.

Additionally, we examine the effect of validation data size by comparing BLEU-2 scores under varying data volumes. Scores generally increase with more data—for instance, with Tool Number = 2, BLEU-2 improves from 0.79 (data = 5) to 0.89 (data = 10).

Tool Number	Sample Size of Validation Data								AVG
	5		10		20		30		
	Layer	BLEU-2	Layer	BLEU-2	Layer	BLEU-2	Layer	BLEU-2	
2	0.6	0.79	2.7	0.89	3.0	0.89	3.2	0.87	0.86
3	0.4	0.81	2.5	0.87	2.8	0.86	3.0	0.85	0.85
4	0.6	0.80	2.8	0.88	3.0	0.87	3.2	0.86	0.86

Table 5: Comparison of the performance of different tools on the text-based molecule design task under different validation data sizes. “Layer” represents the average number of layers of agent composite Tools.

5.2 Validation cost over the four tasks

We validated the composite structure bottom-up, akin to a greedy search. The time and token cost during validation was measured across four tasks. As illustrated in Fig. 3, the cost exhibits near-linear growth within the first three layers (Performance

stabilizes post-convergence in our tasks), transitioning to a steep increase beyond the fourth layer.

5.3 Layer Depth - Performance Scaling

This section investigates the impact of layer depth on performance in the text-based molecule design task. As shown in Tab. 7, increasing the number of stacking layers initially improves BLEU-2 scores, with performance peaking at seven layers. Shallower architectures (e.g., 1–2 layers) may lack the capacity to model complex feature interactions, while deeper configurations beyond seven layers yield diminishing or even negative returns. This suggests a performance plateau likely caused by redundant information or overfitting in overly deep structures.

5.4 Why does the agent composite tool work?

From the case studies, we identify four distinct behavior patterns—correct, modify, judge and reserve, with detailed information on other cases provided in supplementary material. These findings demonstrate the agent-composite tool’s capability to aggregate information from diverse tool structures while simultaneously filtering erroneous outputs. In the text-based molecular design task, initially, Tool A predicts the SMILES string “C1(C)CC1” for cyclopropane, which is incorrect due to a misinterpretation of the three-membered ring, introducing an erroneous branch. To correct this, Tool B is invoked with the query “Cyclopropane” and returns the correct SMILES “C1CC1”, consistent with the standard structure.

Text-based Molecule Design (BLEU-2) - w/o-Tool									
NUM	Chain	Random	FullConnected	Layered	Star	Debate	Avg	Avg_all_tokens	Avg_Time
0	0.548	0.548	0.548	0.548	0.548	0.548	0.548	890.19	8.254
1	0.604	0.604	0.604	0.604	0.604	0.604	0.604	2315.22	15.826
2	0.587	0.587	0.587	0.587	0.587	0.587	0.587	3619.54	21.034
4	0.608	0.642	0.672	0.666	0.661	0.639	0.648	52010.75	74.029
8	0.633	0.701	0.684	0.640	0.649	0.636	0.657	344604.597	227.012
Text-based Molecule Design (BLEU-2) - Tool (Name2SMILES,ChemDFM)									
NUM	Chain	Random	FullConnected	Layered	Star	Debate	Avg	Avg_all_tokens	Avg_Time
0	0.832	0.832	0.832	0.832	0.832	0.832	0.832	2528.79	21.382
1	0.842	0.842	0.842	0.842	0.842	0.842	0.842	3125.64	30.237
2	0.844	0.844	0.844	0.844	0.844	0.844	0.844	18446.91	108.427
4	0.864	0.888	0.885	0.867	0.884	0.889	0.880	150544.061	2207.641
8	0.841	0.890	0.854	0.887	0.891	0.874	0.873	733540.998	6744.788
NUM	Our Agent Composite Tool					Score	Avg_all_tokens	Avg_Time	
2	['ChemDFM_2']					0.898	2801.93	39.103	
3	['Name2SMILES_1','ChemDFM_1']					0.918	2821.38	63.795	
4	['Name2SMILES_3','ChemDFM_0']					0.904	2745.07	78.144	
4	[['ChemDFM_0', 'Name2SMILES_1'], 'ChemDFM_1']					0.925	2851.43	72.484	
8	[['ChemDFM_1','Name2SMILES_1'], ['ChemDFM_1','Name2SMILES_2']]					0.907	1830.11	95.623	

Table 6: Inference Cost vs. Multi-Agent Networks. Each tool’s annotated depth indicates its layer depth after Stage 1. Name2SMILES is a basic retrieve tool for chemistry online database widely used in chemistry agent like ChemCrow.

Layer	1	2	3	4	7	8	10	10+	AVG
BLEU-2	0.88	0.90	0.92	0.93	0.90	0.92	0.91	0.90	0.91

Table 7: Comparison of different agent stacking levels on the performance of text-based molecule design tasks.

5.5 Inference Cost vs. Multi-Agent Networks

Comparison of Multi-Agent Systems and Tool-Augmented LLMs Multi-agent systems (MAS) based on LLMs and our method share some characteristics, including task decomposition, tool invocation, and information sharing. Our study evaluates the performance of six baseline MAS with varied communication architectures against our method in a text-based molecular design task. In Figure 6, our method achieves substantial efficiency improvements over baseline multi-agent networks. In a four-agent configuration, it reduces token consumption to one-eighteenth of the average for MAS while preserving comparable inference latency. This efficiency stems from our method’s optimized coordination of specialized tools, enabling robust performance with minimal computational overhead. Further exploration of these coordina-

tion mechanisms could enhance their applicability to other complex scientific tasks. Detailed descriptions of the MAS configurations are available in the Appendix.

6 Conclusion

We introduced tool amplification for chemistry. It is a novel paradigm that transcends tool orchestration by dynamically amplifying atomic tools’ capabilities through hierarchical coordination within individual tasks. Instantiating this paradigm, ChemAmp leverages bi-phase iterative encapsulation to construct task-specialized composable super-agents from atomic tools (e.g., UniMol2, Chemformer). Evaluations across molecular design, captioning, reaction prediction, and property prediction demonstrate state-of-the-art results, outperforming chemistry specialized models, generalist LLMs, and orchestration-based agents. Critically, ChemAmp achieves these gains with minimal data (≤ 10 samples) while reducing inference token costs by 94% versus vanilla multi-agent systems. This work establishes tool amplification as a scalable, efficient foundation for next-generation LLM-agent systems in complex scientific domains.

Limitations

While ChemAmp demonstrates significant advancements in tool amplification for chemistry tasks, its evaluation has primarily focused on four core tasks: molecular design, molecule captioning, reaction prediction, and property prediction. These tasks, though representative, do not encompass the full spectrum of challenges in chemistry or other scientific domains. Extending tool amplification to additional tasks, such as retrosynthetic planning or multi-step reaction optimization, could reveal further opportunities to refine the bi-phase iterative encapsulation process and enhance its robustness across diverse problem spaces.

Additionally, ChemAmp’s reliance on a small number of samples (≤ 10) for constructing task-specialized super-agents, while efficient, assumes high-quality input data tailored to specific tasks. In scenarios where data is noisy, incomplete, or highly heterogeneous, the framework’s performance may require further adaptation, such as integrating advanced data preprocessing or transfer learning techniques. Future work could explore these adaptations to broaden the applicability of tool amplification in real-world scientific workflows with varying data constraints.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 82394432, and 92249302), and the Shanghai Municipal Science and Technology Major Project (Grant No. 2023SHZDZX02).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldasari, Andrew D White, and Philippe Schwaller. 2023. Chemcrow: Augmenting large-language models with chemistry tools. *Preprint*, arXiv:2304.05376.

Huajun Chen, Keyan Ding, Jing Yu, Junjie Huang, Yuchen Yang, and Qiang Zhang. 2025. Scitoolagent: A knowledge graph-driven scientific agent for multi-tool integration.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.

et al. Guo, Daya. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Preprint*, arXiv:2305.18365.

Taicheng Guo, Changsheng Ma, Xiuying Chen, Bozhao Nan, Kehan Guo, Shichao Pei, Lu Yu, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Modeling non-uniform uncertainty in reaction prediction via boosting and dropout.

Zhichun Guo, Chuxu Zhang, Wenhao Yu, John Herr, Olaf Wiest, Meng Jiang, and Nitesh V. Chawla. 2021. Few-shot graph learning for molecular property prediction. In *Proceedings of the Web Conference 2021*, WWW ’21. ACM.

Rishin Haldar and Debajyoti Mukhopadhyay. 2011. Levenshtein distance technique in dictionary lookup methods: An improved approach. *Preprint*, arXiv:1101.1232.

Yang Han, Ziping Wan, Lu Chen, Kai Yu, and Xin Chen. 2024. From generalist to specialist: A survey of large language models for chemistry. *Preprint*, arXiv:2412.19994.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. 3(1):015022. Publisher: IOP Publishing.

Xiaohong Ji, Zhen Wang, Zhifeng Gao, Hang Zheng, Linfeng Zhang, Guolin Ke, and Weinan E. 2024a. Uni-mol2: Exploring molecular pretraining model at scale. *Preprint*, arXiv:2406.14969.

Xiaohong Ji, Zhen Wang, Zhifeng Gao, Hang Zheng, Linfeng Zhang, Guolin Ke, et al. 2024b. Uni-mol2: Exploring molecular pretraining model at scale. *arXiv preprint arXiv:2406.14969*.

Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi Jaakkola. 2017. Predicting organic reaction outcomes with weisfeiler-lehman network. *Preprint*, arXiv:1709.04555.

Pengfei Liu, Jun Tao, and Zhixiang Ren. 2025. A quantitative analysis of knowledge-learning preferences in large language models in molecular science. *Preprint*, arXiv:2402.04119.

Harry L. Morgan. 1965. The generation of a unique machine description for chemical structures—a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5:107–113.

- Emmanuel Noutahi, Cristian Gabellini, Michael Craig, Jonathan S. C Lim, and Prudencio Tossou. 2023. [Gotta be safe: A new framework for molecular design](#). *Preprint*, arXiv:2310.10773.
- OpenAI and : Aaron Hurst et al. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Siru Ouyang, Zhuosheng Zhang, Bing Yan, Xuan Liu, Yejin Choi, Jiawei Han, and Lianhui Qin. 2024. [Structured chemistry reasoning with large language models](#). *Preprint*, arXiv:2311.09656.
- Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2024. [Scaling large-language-model-based multi-agent collaboration](#). *Preprint*, arXiv:2406.07155.
- Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. 2019. [Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction](#). *ACS Central Science*, 5(9):1572–1583.
- Yaorui Shi, An Zhang, Enzhi Zhang, Zhiyuan Liu, and Xiang Wang. 2023. [ReIm: Leveraging language models for enhanced chemical reaction prediction](#). *Preprint*, arXiv:2310.13590.
- Sakhinana Sagar Srinivas and Venkataramana Runkana. 2024. [Cross-modal learning for chemistry property prediction: Large language models meet graph machine learning](#). *Preprint*, arXiv:2408.14964.
- Xiangru Tang, Tianyu Hu, Muyang Ye, Yanjun Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou, Pan Lu, Zhuosheng Zhang, Yilun Zhao, Arman Cohan, and Mark Gerstein. 2025. [Chemagent: Self-updating library in large language models improves chemical reasoning](#). *Preprint*, arXiv:2501.06590.
- T.T. Tanimoto. 1958. *An Elementary Mathematical Theory of Classification and Prediction*. International Business Machines Corporation.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *arXiv preprint arXiv:2307.09288*.
- Hongwei Wang, Weijiang Li, Xiaomeng Jin, Kyunghyun Cho, Heng Ji, Jiawei Han, and Martin D. Burke. 2021. [Chemical-reaction-aware molecule representation learning](#). *Preprint*, arXiv:2109.09888.
- Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. 2018. [Moleculenet: A benchmark for molecular machine learning](#). *Preprint*, arXiv:1703.00564.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). *Preprint*, arXiv:2210.03629.
- Botao Yu, Frazier N. Baker, Ziru Chen, Garrett Herb, Boyu Gou, Daniel Adu-Ampratwum, Xia Ning, and Huan Sun. 2025. [Chemtoolagent: The impact of tools on language agents for chemistry problem solving](#). *Preprint*, arXiv:2411.07228.
- Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. 2025. [Multi-agent architecture search via agentic supernet](#). *arXiv preprint arXiv:2502.04180*.
- Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024a. [Cut the crap: An economical communication pipeline for llm-based multi-agent systems](#). *arXiv preprint arXiv:2410.02506*.
- Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024b. [Cut the crap: An economical communication pipeline for llm-based multi-agent systems](#). *Preprint*, arXiv:2410.02506.
- Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. 2024c. [Aflow: Automating agentic workflow generation](#). *arXiv preprint arXiv:2410.10762*.
- Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, Kai Yu, and Xin Chen. 2024. [Chemdfm: A large language foundation model for chemistry](#). *Preprint*, arXiv:2401.14818.
- Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. [Gptswarm: Language agents as optimizable graphs](#). In *Forty-first International Conference on Machine Learning*.

Appendix

A ChemAmp Agent Details

Now we will present detailed results and scores for each agent in the Tab. 8, along with the corresponding prompts:

A.1 Agent

A.1.1 Prompt

Our Agent framework is based on the ReAct method to implement tool and reasoning processes.

A.1.2 Naming Rule

To facilitate the comprehension of ChemAmp, we propose a systematic hierarchical naming rules. In addition, in order to prevent the tool name from affecting the agent call, we choose to anonymously process the agent tool, that is, {task name}_{num}.

- **Atomic-to-Composite Amplification:** Hierarchical proxies are constructed through a recursive generation strategy, with the depth of the hierarchy dynamically extendable via the numerical suffix. For instance, "[Name2SMILES_0]" denotes the base tool, while "[Name2SMILES_1]" signifies a first-layer tool, referred to as an Agent Tool (which encapsulates both the tool and the Agent into a new tool) and "[ChemDFM_2]".
- **Cross-Composite Synergy:** The combination of multiple tools within an Agent is represented in a list format, utilizing depth-first traversal to generate sub-tools, thereby forming the final toolset for the agent. For example, the structure "[Name2SMILES_1", "ChemDFM_2]" represents a flat structure with tools at the same level ([A, B, ...]), while the structure "[[Name2SMILES_0', 'ChemDFM_1'], 'Name2SMILES_1', 'ChemDFM_0']" illustrates a nested structure ([A, B], C, D)), where tools A and B are first combined before being integrated with tool C and D.

A.1.3 Dataset Setting

A.2 Text-based Molecule Design

A.2.1 Task Introduction

The test set of ChEBI-20-MM is exploited for this task in ChemLLMBench. Models are asked to predict the SMILES of the molecule that fits the given description. Considering the low accuracy of the

models, we use BLEU-2 as the training metric and use metrics such as Exact, Dis and others during the testing stage.

A.3 Text-based Molecule Design

A.3.1 Task Introduction

The test set of ChEBI-20-MM is exploited for this task in ChemLLMBench. Models are asked to predict the SMILES of the molecule that fits the given description. Considering the low accuracy of the models, we use BLEU-2 as the training metric and use metrics such as Exact, Dis and others during the testing stage.

A.3.2 Prompt

We use a simpler prompt compared with the prompt introduced in (Guo et al., 2023)

Prompt: Text-based Molecule Design

You are an expert chemist. Given the molecular requirements description, your task is to design a new molecule SMILES:
Molecular requirements description::

A.4 Molecule Captioning

A.4.1 Task Introduction

The test set is the same with the Text-based Molecule Design task. Because this is the mirroring task, which generates a detailed description by giving a SMILES to the models. In this task, we also choose the BLEU-2 as the metric in the training stage. When in the test stage, more metrics, like BLEU and ROUGE, are utilized to Measure the performance of the model.

A.4.2 Prompt

We also use a simpler prompt compared with the prompt introduced in (Guo et al., 2023)

Prompt: Molecule Captioning

You are an expert chemist. Given the molecular SMILES, your task is to provide the detailed description((The molecule is ...) of the molecule.
Please strictly follow the format, no other information can be provided.
Molecular SMILES:

Task	Validation		Result
	Final tool	score	
Text-based Molecule Design	['Name2SMILES_3', 'ChemDFM_0']	0.80	0.90
	[['ChemDFM_0', 'Name2SMILES_1', 'ChemDFM_1']]	0.81	0.93
	[['ChemDFM_1', 'Name2SMILES_1'], ['ChemDFM_1', 'Name2SMILES_2']]	0.91	0.91
Molecule Captioning	['SMILES2Description_2', 'TextChemT5_0']	0.79	0.73
	['SMILES2Description_3']	0.70	0.65
	['TextChemT5_1', 'SMILES2Description_1']	0.71	0.66
Reaction Prediction	['Chemformer_0', 'SMILES2Property_2']	1.00	0.90
	['Chemformer_2']	0.90	0.89
	['Chemformer_1', 'SMILES2Property_1']	0.90	0.85
Property Prediction (BACE)	['UniMol_1', 'SMILES2Property_1']	0.80	72.1
	[['UniMol_1', 'SMILES2Property_0', 'SMILES2Property_0']]	0.82	81.4
	['UniMol_3']	0.75	78.6
Property Prediction (BBBP)	['UniMol_1', 'SMILES2Property_2']	0.68	71.1
	['SMILES2Property_3']	0.75	69.3
	['UniMol_2']	0.73	70.6
Property Prediction (Clintox)	[['UniMol_1', 'SMILES2Property_0', 'UniMol_1', 'SMILES2Property_1']]	0.70	72.3
	['SMILES2Property_2']	0.68	61.4
	['UniMol_1', 'SMILES2Property_1']	0.65	69.1
Property Prediction (HIV)	[['SMILES2Property_1', 'UniMol_0', 'SMILES2Property_1']]	1.00	97.4
	['UniMol_0', 'SMILES2Property_1']	0.85	90.1
	['UniMol_2']	0.90	96.9
Property Prediction (Tox21)	['UniMol_2']	0.78	92.3
	['UniMol_0', 'SMILES2Property_2']	0.80	79.6
	['SMILES2Property_2']	0.85	74.9

Table 8: Stacking results of different tasks. The **bold** font represents the most suitable toolset obtained in the task experiment.

Ability	Task	Task Type	Dataset	#val	#test
Understanding	Molecular Property Prediction	Classification	BBBP, HIV, BACE, Tox21, ClinTox	250	100
Reasoning	Reaction Prediction	Generation	USPTO-MIT	100	100
Reasoning	Text-Based Molecule Design	Generation	ChEBI-20-MM	100	100
Explaining	Molecule Captioning	Generation	ChEBI-20-MM	100	100

Table 9: Details of the training and test sets for the four chemistry tasks.

A.5 Molecular Property Prediction

A.5.1 Task Introduction

The molecular property prediction tasks in ChemLLMBench consist of five tasks from MoleculeNet benchmark (Wu et al., 2018), including BACE, BBBP, HIV, ClinTox, and Tox21. Among these, BACE and BBBP are balanced binary classification tasks, while HIV represents an unbalanced binary classification task. ClinTox consists of 2 unbalanced binary classification tasks, and Tox21 comprises 21 unbalanced binary classification tasks. In this task, we choose the AUC-ROC as the first metric in the training stage. Considering the calculation method of AUC-ROC for large language models, we also introduced Accuracy as a second metric for test stage.

A.5.2 Prompt

We use the same prompts introduced in (Guo et al., 2023)

A.6 Reaction Prediction

A.6.1 Task Introduction

The reaction prediction task asks the model to predict the product of the given reaction. ChemLLMBench utilizes the USPTO-MIT dataset for this task. Since the benchmark metric is Accuracy, we also chose Accuracy as the training metric, and considering that the answer is also SMILES, we adopted the same metric as Molecular Design task for measurement during the testing stage.

A.6.2 Prompt

We reformat the prompt provided (Guo et al., 2023).

Prompt: Reaction Prediction

Given an incomplete chemical reaction equation in SMILES notation (format: reactants»product, where multiple reactants are separated by dots '.'), predict and complete the missing products marked as '___'. The response should only contain the only one SMILES representation of the missing molecule, without any additional explanation (Note: Please only output only one final product). Please answer the question based on the following Chemical reaction equation:

B Multi-agent Implementation Details

In this section, we will discuss how to implement multi-agent systems and specific ways of information transmission, including chain, random, star, full-connected, layered, and debate graphs.

B.1 Framework

In order to complete chemical tasks, we divided multi-agent into two types: agents with tools and agents without tools, and tested them on the first task, the Text-based Molecule Design task. The overall of our multi-agent framework is a modification of the framework of (Zhang et al., 2024b) and (Qian et al., 2024) that utilized different spatial and temporal masks to complete in the following six multi-agent structures: Chain, Random, FullConnected, Layered, Star and Debate mode.

However, since we are modifying their approach with a greater focus on information transmission and are also limited by API calls, we can only make a one-sided comparison regarding the recording of tokens and time.

B.2 Implementation Details

In multi-agent systems, information transmission is a critical factor for enhancing performance. By utilizing various structures, information can be conveyed through multiple pathways. To improve the efficiency of information collection, we have adopted the Final decision approach. Specifically, at the end of all structures, we have integrated a FinalRefer Agent to perform the final summary and decision-making. The FinalRefer prompt is followed:

Prompt: FinalRefer

You are a strategic planning and final integration agent. You will be given a graduate-level question and reasoning outputs from all other agents. Your task is to integrate all the information into a single, cohesive answer with detailed reasoning and evidence. Your final output should: 1. Summarize the contributions from all agents, highlighting key insights.

3. Provide the final answer with a clear and detailed explanation.

4. Conclude with the final answer on a new line with the format: "The final answer is 'SMILES'"

Here is the question:question. At the same time, the output of other agents is as follows:

answers

In the implementation of the agents with tools, we modified all agents along the path except for the Final agent, while still following the ReAct framework for tool calling. During this process, due to the constraints of API calls, both the time required and the number of tokens used will be greater compared to agents without tools.

B.3 Spatial Communication Topologies

B.3.1 Chain

The chain graph (Fig. 4) is one of the most widely utilized communication architectures in contemporary multi-agent systems. In this architecture, the first agent receives input from the user, transforms it into new instruction, and subsequently forwards it to the next agent. Generally, the final agent in the chain provides a summary and answers.

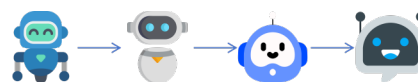


Figure 4: Demonstration of **chain** structure

B.3.2 Random

The random graph refers to a sparse graph randomly sampled from a complete graph, as shown in the Fig. 5. They will execute asynchronously in multiple rounds and then randomly transmit information to the target agent. Finally, all the answers and information will transmit to the Final agent to make a final answer.

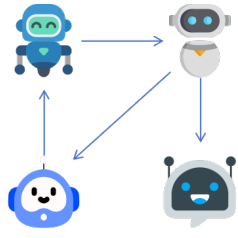


Figure 5: Demonstration of **random** structure

B.3.3 FullConnected

The fullconnected graph (Fig. 6) is a directed graphs compared to traditional fully linked undirected graphs, which transmit information in a certain order to complete this topology structure. The final agent summarizes the dialogue and provides a concluding output or reflection.

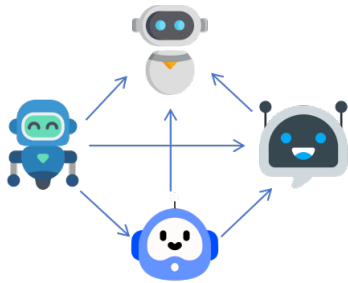


Figure 6: Demonstration of **FullConnected** structure

B.3.4 Layered

The layered graph (Fig. 7, (Qian et al., 2024)) refers to a stacked configuration similar to a multilayer perceptron (MLP). The first layer agents will feed to the agents in the second layer, and the final layer will make the summary and final-decision.

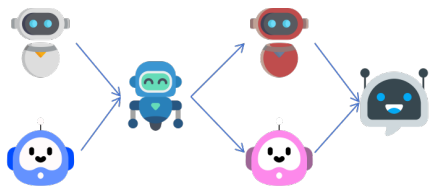


Figure 7: Demonstration of **Layered** structure

B.3.5 Star

The star graph (Fig. 8) resembles the tree structure. Firstly, the problem will be handed over to the external leaf nodes for processing, and the obtained answer will be passed to the central root node, which will be repeated multiple times. Finally, the root node will give a summary and make the decision.

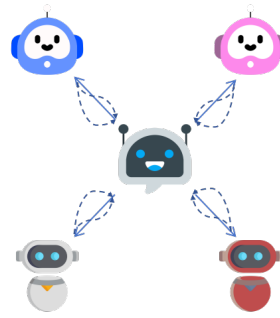


Figure 8: Demonstration of **Star** structure

B.3.6 Debate

The debate graph (Fig. 9, (Du et al., 2023)) is multiple agents to engage in a debate, where in each round, every agent receives the outputs of all agents from the previous round before making their own statements. Generally, the finalRefer agent will help them to make the final decision.

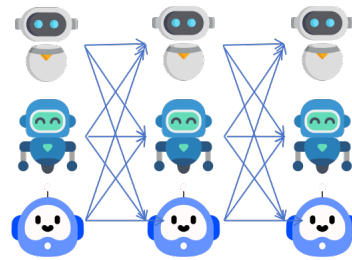


Figure 9: Demonstration of **Debate** structure

B.3.7 Tree-like&Ours

In order to better understand our composite tool structure, we will compare it with a tree like multi-agent pipeline that is relatively similar. The tree graph usually has the root node as the manager to supervise the nodes below to complete various tasks, and finally return the results of the leaf nodes to the root node for processing. Overall, this is just a top-down process of information transmission. As shown in the Fig. 10

As for our composite tool structure, from the perspective of information transmission, the main agent of the root node also receives the information completed from below, but there is a difference. For the so-called tool node, it is a bottom-up process. After continuous information superposition step by step, it is passed upward through the parent node and then given to the main agent for processing. It can be seen from the figure that in each transmission process, whether it is the root node or the parent node, they all selectively accept the information from the child node, and it is not like a

tree structure that is passed downward.

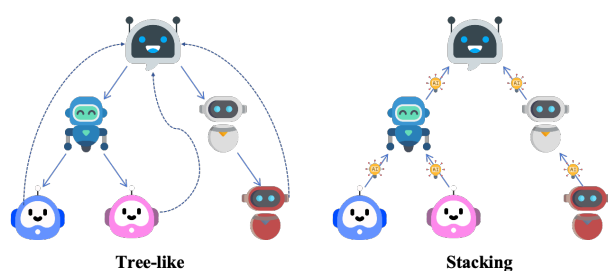


Figure 10: Demonstration of **Tree** and **Ours** structure. In the composite tool structure, the icon '☀️' means the parent node can choose which child node's information to accept.

C Case Study

There are four distinct composite tool behavior patterns-**correct**, **modify**, **judge** and **reserve** that make the composite tool works. Here are three other cases for these patterns in the Tab. 10:

C.1 Correct

In the text-based molecular design task, the model uses two callable tools. Initially, Tool A predicts the SMILES string "C1(C)CC1" for cyclopropane, which is incorrect due to a misinterpretation of the three-membered ring, introducing an erroneous branch. To correct this, Tool B is invoked with the query "Cyclopropane" and returns the correct SMILES "C1CC1", consistent with the standard structure.

C.2 Modify

As shown in the table, this is a common approach to using various tools. First, the problem is decomposed, and the RAG tool is used to retrieve information on each sub-question to obtain a preliminary answer. Then, subsequent processing is carried out using computational tools. Alternatively, one can first obtain an answer through computational tools, then have the agent self-assess the correctness of that answer, and finally use the RAG tool for cross-verify, thereby refining the answer and improving accuracy.

C.3 Judge

Judge refers to the process of selecting between two candidate answers based on the model's knowledge in chemistry. This usually happens when two agent tools are available. When confronted with two anonymous tools that have the same descriptions, the model often opts to call both tools simultaneously before making a judgment. Furthermore,

when using GPT-4o as the agent model, it typically demonstrates excellent judgment abilities.

C.4 Reserve

Reserve is a relatively uncommon phenomenon for large language models. Due to the hallucination problem, they usually choose to cover up knowledge blind spots to generate answers. However, when faced with multiple tools, the agent faces several answers. After thinking, it does not judge which one is more suitable, so it chooses to reserve opinions. Although this will reduce accuracy, in some ways it is a good thing to emerge this ability.

Agent Behavior Pattern	Input	Output
Correct	<p>The molecule is a cycloalkane composed of a three-carbon ring. It functions as an inhalation anesthetic. It is a cycloalkane and a member of the cyclopropane class. What is its predicted SMILES?</p> <p>Gold Answer: C1CC1</p>	<p>According to the provided description, the molecule is cyclopropane, which is a cycloalkane formed by three carbon atoms in a ring, well known for its use as an inhalation anesthetic. To confirm the SMILES notation of cyclopropane, I used the tool-A.</p> <p>Action: Call tool-A</p> <p>Input: What is the SMILES notation for cyclopropane (a three-carbon cycloalkane used as an inhalation anesthetic)?</p> <p>Output: CC1(C)CC1</p> <p>Observation: The returned "CC1(C)CC1" does not correspond to cyclopropane. Cyclopropane should be a simple three-carbon ring, typically represented as "C1CC1". Next, I will call the tool-B to further correct the result.</p> <p>Action: Call tool-B</p> <p>Input: Cyclopropane</p> <p>Output: C1CC1</p> <p>Observation: Based on the known structure of cyclopropane, the result from this tool is correct. Therefore, I will provide the correct SMILES based on the understanding of the chemical structure.</p> <p>Final Answer: C1CC1</p>
Modify	<p>The molecule is a peptide anion that is the conjugate base of gamma-Glu-Abu, obtained by removal of protons from the two carboxy groups as well as protonation of the amino group; major species at pH 7.3. It is a conjugate base of a gamma-Glu-Abu. Please try to infer the SMILES of this molecule.</p> <p>Gold Answer: CCC(NC(=O)CCC([NH3+])C(=O)[O-])C(=O)[O-]</p>	<p>To find the SMILES representation of the peptide anion that is the conjugate base of gamma-Glu-Abu, I first need to understand the structure of gamma-Glu-Abu. Gamma-Glu-Abu stands for gamma-glutamyl-aminobutyric acid. I will start by finding the SMILES of gamma-Glu-Abu.</p> <p>Action: Call tool-B</p> <p>Input: gamma-Glu-Abu</p> <p>Output: CCC(NC(=O)CCC(N)C(=O)O)C(=O)O</p> <p>Observation: The next step is to modify this SMILES to represent the conjugate base of gamma-Glu-Abu at pH 7.3, where the two carboxyl groups are deprotonated and the amino group is protonated. This involves changing the carboxyl groups from "-C(=O)O" to "-C(=O)[O-]" and the amino group from "-N" to "-[NH3+]".</p> <p>Action: Call tool-A</p> <p>Input: Modify the "CCC(NC(=O)CCC([NH3+])C(=O)[O-])C(=O)[O-]" from "-C(=O)O" to "-C(=O)[O-]" and the amino group from "-N" to "-[NH3+]".</p> <p>Output: CCC(NC(=O)CCC([NH3+])C(=O)[O-])C(=O)[O-]</p> <p>Thought: I now know the final answer</p> <p>Final Answer: CCC(NC(=O)CCC([NH3+])C(=O)[O-])C(=O)[O-]</p>
Judge	<p>The molecule is a cycloalkane composed of three carbon atoms to form a ring. It has a role as an inhalation anaesthetic. It is a cycloalkane and a member of cyclopropanes. Please try to infer the SMILES of this molecule.</p> <p>Gold Answer: CC(=O)N[C@@H]1[C@@H](O)[C@@H]2O[C@@H](C)[C@@H](O)[C@@H](O)[C@@H]2O[C@@H](O)[C@@H](O)[C@@H](O)[C@@H]1O</p>	<p>The molecule described is a complex carbohydrate structure involving alpha-L-Fucp, beta-D-Galp, and D-GlcNAc units. The specific linkages and configurations are crucial for determining the correct structure. Given the complexity of the molecule, the best approach is to use the Molecule_Design tools, which are designed to handle detailed questions about molecular structures. I will start with Molecule_Design_1 to see if it can provide a correct SMILES representation based on the detailed description provided.</p> <p>Action: Call tool-C</p> <p>Action Input: The molecule is an alpha-L-Fucp-(1->2)-beta-D-Galp-(1->3)-[alpha-L-Fucp-(1->4)]-D-GlcNAc where the glucosamine at the reducing end has beta-configuration at its anomeric centre. It has a role as an epitope. Please provide the SMILES of this molecule.</p> <p>Observation: CC(=O)N[C@@H]1[C@@H](O)[C@@H]2O[C@@H](C)[C@@H](O)[C@@H](O)[C@@H]2O[C@@H](O)[C@@H](O)[C@@H](O)[C@@H]1O</p> <p>Thought: The SMILES provided by tool-C seems to represent a complex carbohydrate structure, but to ensure accuracy, I will use tool-D for a second opinion. This will help verify the correctness of the SMILES or provide an alternative if there are discrepancies.</p> <p>Action: Call tool-D</p> <p>Action Input: The molecule is an alpha-L-Fucp-(1->2)-beta-D-Galp-(1->3)-[alpha-L-Fucp-(1->4)]-D-GlcNAc where the glucosamine at the reducing end has beta-configuration at its anomeric centre. It has a role as an epitope. Please provide the SMILES of this molecule.</p> <p>Observation: Nc1ccn([C@@H]2O[C@@H](COP(=O)([O-])[O-])[C@@H](O)[C@@H]2O)c(=O)n1</p> <p>Thought: The SMILES provided by tool-D is significantly different from the one provided by tool-C, indicating a discrepancy. The structure from tool-D seems less likely to represent the complex carbohydrate structure described in the question. The SMILES from tool-C appears more plausible as it includes multiple sugar units linked together, which aligns better with the description of the molecule involving alpha-L-Fucp, beta-D-Galp, and D-GlcNAc. Therefore, I will consider the SMILES from tool-C as the more likely correct representation.</p> <p>Thought: I now know the final answer</p> <p>Final Answer: CC(=O)N[C@@H]1[C@@H](O)[C@@H]2O[C@@H](C)[C@@H](O)[C@@H](O)[C@@H]2O[C@@H](O)[C@@H](O)[C@@H]1O</p>
Reserve	<p>The molecule is a linear amino tetrasaccharide composed of N-acetyl-beta-D-galactosamine, alpha-D-galactose, beta-D-galactose and N-acetyl-beta-D-galactosamine units joined by sequential (1->3)-, (1->4)- and (1->4)-linkages. It has a role as an epitope. Please try to give SMILES of this molecule.</p> <p>Gold Answer: CC(=O)N[C@@H]1[C@@H](O)[C@@H](O)[C@@H]2O[C@@H](CO)[C@@H](O)[C@@H]3O[C@@H](CO)[C@@H](O)[C@@H]4O[C@@H](CO)[C@@H](O)[C@@H]4NC(C)=O[C@@H]3O[C@@H](O)[C@@H]2O[C@@H](CO)O[C@@H]1O</p>	<p>The SMILES representation for the molecule described is not readily available due to the complexity of the glycosidic linkages and the specific stereochemistry involved. Further detailed chemical synthesis knowledge or specialized software would be required to accurately construct the full SMILES notation of this tetrasaccharide.</p> <p>Thought: The initial attempt using the Molecule_Design_2 tool did not yield a direct SMILES result, likely due to the complexity of the molecule described. As an alternative approach, I can attempt to construct the molecule step-by-step using the tool-B tool for each individual sugar unit and then logically deduce the possible structure of the entire molecule. However, constructing the full SMILES for the entire molecule including specific linkages manually might not be feasible without specialized chemical software. I will first retrieve the SMILES for each sugar unit.</p> <p>Action: Call tool-B</p> <p>Action Input: N-acetyl-beta-D-galactosamine</p> <p>Observation: CC(=O)N[C@@H]1[C@@H](O)[C@@H](O)[C@@H](CO)O[C@@H]1O</p> <p>Thought: I have obtained the SMILES for N-acetyl-beta-D-galactosamine. Next, I will retrieve the SMILES for alpha-D-galactose.</p> <p>Action: Call tool-B</p> <p>Action Input: alpha-D-galactose</p> <p>Observation: OC[C@@H]1O[C@@H](O)[C@@H](O)[C@@H](O)[C@@H]1O</p> <p>Thought: I have obtained the SMILES for alpha-D-galactose. Next, I will retrieve the SMILES for beta-D-galactose.</p> <p>Action: Call tool-A</p> <p>Action Input: What is the SMILES of beta-D-galactose</p> <p>Observation: OC[C@@H]1O[C@@H](O)[C@@H](O)[C@@H](O)[C@@H]1O</p> <p>Thought: I have obtained the SMILES for beta-D-galactose. Since the molecule starts and ends with N-acetyl-beta-D-galactosamine, and I already have the SMILES for this unit, I do not need to retrieve it again. However, constructing the full SMILES string for the tetrasaccharide with specific glycosidic linkages (1->3, 1->4, 1->4) between these units is beyond the capability of the Call tool-B tool and requires specialized chemical synthesis software or expertise in carbohydrate chemistry. Therefore, I will conclude my process.</p> <p>Thought: I now know the final answer</p> <p>Final Answer: Unable to provide the exact SMILES string for the described tetrasaccharide due to the complexity of the glycosidic linkages and the limitations of the tools available.</p>

Table 10: Case analysis of some behavior patterns of agents.