TOOLING OR NOT TOOLING? THE IMPACT OF TOOLS ON LANGUAGE AGENTS FOR CHEMISTRY PROBLEM SOLVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown promise in various domains but face challenges in chemistry due to limited domain knowledge and computational capabilities. To address these issues, tool-augmented language agents like ChemCrow and Coscientist have been developed. However, their evaluations remain narrow in scope, leaving an unclear understanding of how these tool-augmented agents perform across various real-world applications. In this study, we conduct a comprehensive evaluation to bridge this gap. Specifically, we develop ChemAgent, the most capable chemistry agent to date, equipped with 29 tools capable of handling a wide spectrum of tasks. We then conduct a comprehensive assessment across three datasets, namely SMolInstruct, MMLU-chemistry, and GPQA-chemistry, which can be categorized into specialized chemistry tasks and general chemistry questions. Surprisingly, tool-augmented agents do not consistently outperform the base LLM without tools, and the impact of tool augmentation is highly task-dependent: It provides substantial gains in specialized chemistry tasks but potentially hinders performance in general chemistry questions. We further engage domain experts and conduct error analysis, revealing that errors in general chemistry questions primarily occur due to minor inaccuracies at intermediate stages of the problem-solving process, highlighting the need for further research into balancing tool use with intrinsic reasoning abilities, to maximize the effectiveness of language agents in chemistry. [1]

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable capabilities across multiple domains, showcasing their potential as versatile problem-solving tools. However, when it comes to chemistry, these models face significant challenges, such as incorrect calculation, lack of domain knowledge, or their inability to perform certain tasks like reaction prediction (Ramos et al., 2024; Mirza et al., 2024). To address this limitation, researchers have proposed LLM-based agents integrated with specialized tools to tackle a wide range of chemistry-related problems. For example, ChemCrow (M. Bran et al., 2024) incorporates 18 tools, ranging from web search to chemical reaction prediction, significantly expanding LLMs' capabilities in chemistry. Another notable example is Coscientist (Boiko et al., 2023), which integrates control of a cloud lab, enabling LLMs to automate chemical experiments.

Despite the promise of these tool-augmented LLMs, existing evaluations have been largely qualitative and very limited in scope. For instance, ChemCrow (M. Bran et al., 2024) was assessed using only 14 self-created specific tasks, and they primarily focuse on synthesis of compounds. Similarly, Coscientist's evaluation (Boiko et al., 2023) involved merely six tasks. These narrow assessments leave a significant gap in our understanding of how these tool-augmented agents perform across diverse chemistry tasks in real-world applications.

In this work, we aim to conduct a comprehensive evaluation of tool-augmented agents across diverse chemistry tasks to grasp a deep understanding of the potential and limitations of existing

[1] Our code and data will be released.

agents. Towards this end, we make a series of efforts: (1) We introduce ChemAgent, the most capable chemistry agent to date. It leverages the ReAct framework (Yao et al., 2023), and integrates 29 tools including PubChem and molecular property predictors, as well as many present in ChemCrow (M. Bran et al., 2024). (2) We compile three datasets that cover different types of chemistry problems in real world for comprehensive evaluation: SMolInstruct (Yu et al., 2024), which contains 14 types of specialized molecule-centric tasks; MMLU-chemistry , a subset of the MMLU benchmark (Hendrycks et al., 2021) that contains high school and college exam-like questions; and GPQA-chemistry, a subset of the GPQA benchamrk (Rein et al., 2023) that contains difficult graduate-level questions. (3) In order to conduct a meaningful error analysis with actionable insights, we propose a reasoning-grounding abstraction framework for existing chemistry agents, where reasoning means to refelct current status and plan for next step, and grounding means to ground the plan into doable actions in environment. (4) We engage with chemistry experts to conduct error analysis, where we analyze the error in each sample, so as to understand the places where agents make mistakes.

Through comprehensive experiments, we demonstrate that **ChemAgent substantially outperforms ChemCrow on all chemistry problems.** However, contrary to expectation, **ChemAgent does not consistently outperform the base LLM without tools, and the impact of tool augmentation is highly dependent on task characteristics.** Specifically, for specialized chemistry tasks involving professional molecular representations (e.g., SMILES (Weininger, 1988)) and specialized chemical operations (e.g., compound synthesis, property prediction), augmenting LLMs with task-specific tools can yield substantial performance gains. Conversely, for general chemistry questions that require more extensive internal knowledge and reasoning, there often lacks specific tools to address these needs adequately. In such cases, tool augmentation may potentially impair LLMs' intrinsic reasoning abilities and lead to diminished performance. Further manual analysis with domain experts shows that errors on general chemistry questions primarily occur due to minor inaccuracies at intermediate stages of the problem-solving process, suggesting the need to improve the intrinsic reasoning abilities of tool-augmented LLMs.

2 CHEMAGENT

We present ChemAgent, an LLM-based agent for chemistry tasks. The framework, illustrated in Figure 1, follows the ReAct paradigm (Yao et al., 2023). Upon receiving a user task, the agent iterates through a three-step process: (1) Thought generation, analyzing the current situation and planning subsequent steps; (2) Action determination, selecting the appropriate tool and its input based on the generated thought; and (3) Observation, processing the results or feedback from the environment post-action execution. This iterative cycle of Thought, Action, and Observation continues until task completion or conclusion.

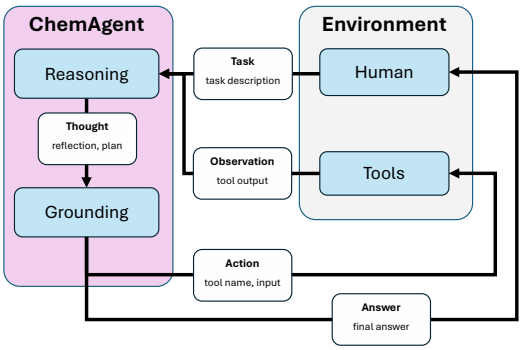


Figure 1: The ChemAgent Framework.

To facilitate our subsequent error analysis, we identify two essential cognitive abilities required in this framework: (1) **Reasoning**: This module serves as the core decision-making unit, responsible for comprehending user queries and tool outputs, assessing current status, and formulating subsequent steps. (2) **Grounding**: Based on the "thought" provided by the reasoning module, this component determines the appropriate tool to execute and its corresponding input. These two abilities are fundamental to the agent workflow and will be further examined in our error analysis.

To enhance ChemAgent's capabilities, we have developed an extensive tool set comprising 29 distinct tools, categorized into general, molecule, and reaction tools. In comparison to ChemCrow (M. Bran et al., 2024), our tool set incorporates additional tools such as PubchemSearchQA, which leverages an LLM to retrieve and extract authorized, comprehensive compound information from PubChem (Kim et al., 2019), and MoleculePropertyPredictor tools, which employ neural network-based models for specific molecular property predictions. We have also improved several tools

present in ChemCrow. For instance, SMILES2Name has been enhanced by integrating multiple information sources, resulting in a more robust service. WebSearch has been upgraded with an LLM-enhanced searching service, providing more comprehensive and organized search results. Notably, we introduce the AiExpert tool, an LLM instructed to answer any questions. It is designed to leverage the LLM's internal knowledge and address scenarios where other tools cannot handle (e.g., for analysis tasks). For a comprehensive overview of the tool set, please refer to Appendix A.

3 EXPERIMENT

Table 1: Datasets used in our experiments.

Category	Dataset	# Sample	Specific task type
Specialized tasks	SMolInstruct	700	molecule-centric tasks
General Questions	MMLU-Chemistry	70	High school- and college-level questions
	GPQA-Chemistry	93	Graduate-level questions

To thoroughly assess models' abilities on various aspects of chemistry, we select three distinct datasets that fall into two categories as listed in Table 1. **Specialized chemistry tasks** focus on practical, experiment-like tasks involving molecular manipulations, predictions, and representations. This category includes SMolInstruct (Yu et al., 2024), which contains multiple types of molecule-centric tasks. **General chemistry questions**, on the other hand, resemble questions appearing in exams at different levels and test a wide range of fundamental knowledge and general reasoning in chemistry. This category includes MMLU-Chemistry, a chemistry subset of the MMLU benchmark (Hendrycks et al., 2021) that consists of high school and college questions, and GPQA-Chemistry, the chemistry section of the GPQA-Diamond benchmark (Rein et al., 2023) that consists of difficult graduate-level questions.

We compare our ChemAgent with two types of methods: (1) The first type comprises state-of-the-art (SoTA) base LLMs, specifically GPT-4o and Claude-3.5-Sonnet, selected for their superior capabilities in chemistry among existing LLMs. (2) ChemCrow (M. Bran et al., 2024), a pioneering chemistry-focused agent equipped with 18 expert-designed tools. For ChemCrow and ChemAgent, we utilize GPT-4o or Claude-3.5-Sonnet as the backbone language model[2]. For brevity, we refer to these foundational models as GPT and Claude, respectively, unless otherwise specified.

In the following subsections, we will first present the overall performance across all the three datasets (Section 3.1), and then conduct detailed error analysis in Section 3.2.

3.1 OVERALL PERFORMANCE

3.1.1 SMOLINSTRUCT

The SMolInstruct dataset comprises 14 specific chemistry tasks, including forward synthesis, name conversion, and property prediction, etc. (Yu et al., 2024). For evaluation, we randomly select 50 samples from the test set for each task. Following Yu et al. (2024), we use the same set of metrics for the tasks. For reference, we also include SoTA non-LLM models used in Yu et al. (2024) as well as LlaSMol[3], a fine-tuned Mistral model (Jiang et al., 2023) using SMolInstruct. For SoTA non-LLM models and LlaSMol, we adopt their own formats of input and output. For all the other models, we prompt them to think step by step, i.e., using chain-of-thought (CoT), and wrap their final answers with "<ANSWER>" and "</ANSWER>" to facilitate answer extraction.

The results are presented in Table 2 and Table 3. We can draw some key findings as follows:

The tool-augmented ChemAgent models exhibit substantial improvements over their base LLM counterparts. Their performance is comparable to, and in many cases surpasses, that of the SoTA non-LLM models and LlaSMol. This enhancement highlights the critical role of domain-specific tools in augmenting LLMs' capabilities.

[2]Model versions: `gpt-4o-2024-08-06` and `claude-3-5-sonnet-20240620`.

[3]`https://huggingface.co/osunlp/LlaSMol-Mistral-7B`

Table 2: The results on SMolInstruct for name conversion and property prediction tasks. The metrics are adopted from Yu et al. (2024), and all the metrics except RMSE are in percentage.

Model	NC					PP					
	I2F	I2S		S2F	S2I	ESOL	Lipo	BBBP	Clintox	HIV	SIDER
	EM	EM	Valid	EM	EM	RMSE↓	RMSE↓	Acc	Acc	Acc	Acc
SoTA non-LLM models	96.0	**68.0**	**100.0**	**100.0**	54.0	**0.808**	**0.527**	88.0	**90.0**	**94.0**	70.0
GPT-4o	12.0	0.0	66.0	8.0	0.0	1.315	1.264	70.0	36.0	86.0	44.0
Claude-3.5-Sonnet	4.0	10.0	70.0	4.0	2.0	1.443	1.267	78.0	50.0	88.0	62.0
LlaSMol	92.0	60.0	96.0	96.0	34.0	1.062	1.164	82.0	98.0	**94.0**	**74.0**
ChemCrow (GPT)	18.0	10.0	18.0	88.0	2.0	4.376	2.061	46.0	62.0	74.0	36.0
ChemCrow (Claude)	16.0	14.0	18.0	42.0	2.0	2.025	1.179	60.0	34.0	92.0	32.0
ChemAgent (GPT)	**100.0**	64.0	**100.0**	**100.0**	**70.0**	0.812	0.529	**90.0**	82.0	**94.0**	70.0
ChemAgent (Claude)	**100.0**	**68.0**	**100.0**	**100.0**	**70.0**	1.131	0.531	**90.0**	58.0	92.0	68.0

Table 3: The results on SMolInstruct for task MC, MG, FS, and RS. All the metrics are adopted from Yu et al. (2024), and all except METEOR are in percentage.

Model	MC	MG			FS			RS		
	METEOR	EM	FTS	Valid	EM	FTS	Valid	EM	FTS	Valid
SoTA non-LLM models	**0.539**	32.0	75.7	96.0	78.0	91.7	**100.0**	**42.0**	**80.5**	**100.0**
GPT-4o	0.152	10.0	57.5	84.0	12.0	46.3	84.0	0.0	36.0	84.0
Claude-3.5-Sonnet	0.211	12.0	67.5	90.0	22.0	60.9	98.0	0.0	45.7	90.0
LlaSMol	0.426	22.0	67.0	98.0	56.0	83.4	**100.0**	26.0	70.3	100.0
ChemCrow (GPT)	0.195	34.0	79.9	68.0	72.0	**92.5**	92.0	8.0	49.0	74.0
ChemCrow (Claude)	0.255	40.0	81.0	86.0	70.0	90.5	92.0	22.0	0.0	90.0
ChemAgent (GPT)	0.510	28.0	76.8	90.0	78.0	92.1	98.0	**42.0**	78.0	98.0
ChemAgent (Claude)	0.443	**44.0**	**83.5**	**100.0**	**80.0**	92.2	**100.0**	**42.0**	78.6	**100.0**

While Claude-3.5-Sonnet generally outperforms GPT-4o, their performance as ChemAgent backbones is comparable. This parity in performance can be attributed to the nature of the SMolInstruct tasks, which primarily require effective tool utilization rather than extensive knowledge or complex reasoning abilities inherent to the LLMs themselves. Despite differences in tool-use preferences, which lead to varying performance in some tasks, both models demonstrate proficiency as "tool users," effectively leveraging the provided resources to address the given problems.

In comparison to ChemCrow, the existing chemistry-oriented agent equipped with various tools, ChemAgent demonstrates superior performance. Our analysis suggests that this performance disparity may be attributed to ChemCrow's limited tool set and the potential lack of robustness in its tool implementations. For instance, ChemCrow's apparent deficiency in molecular property prediction tools and its limited web search capabilities seem to hinder its performance in property prediction tasks. Conversely, ChemAgent's more comprehensive and robust tool set appears to provide a more holistic information source for LLMs to leverage effectively.

3.1.2 MMLU-CHEMISTRY

To effectively and efficiently evaluate the models, we build MMLU-Chemistry, a subset of 70 chemistry question samples derived from the widely-used MMLU dataset (Hendrycks et al., 2021). Specifically, to increase the difficulty and differentiation of the questions, while avoiding erroneous samples presented in the original MMLU, we select samples that appear in both MMLU-Pro (Wang et al., 2024) and MMLU-Redux (Gema et al., 2024). These two datasets are verified versions of MMLU, and MMLU-Pro has extended the answer options from 4 to 10 to introduce more challenges. When the gold standard answers from both sources match, we utilize the 10 options from MMLU-Pro. In cases of discrepancies, we manually review and correct any potential issues. To reduce the cost of evaluation, we eliminated samples where all models performed correctly in our preliminary experiments. This results in a final set of 70 questions, divided evenly between 35 high school-level and 35 college-level questions.

To understand the effect of few-shot learning, we introduce a 5-shot setting in comparison with 0-shot for the base LLMs and ChemAgent. The questions of the in-context examples are originally

from MMLU's and MMLU-Pro's development set, and we manually construct CoT solutions for the base LLMs and tool-using step-wise solutions for ChemAgent. The order of the examples is randomized for each test sample. All the models are prompted to generate a CoT solution and close the solution with "the answer is ..." to facilitate the answer extraction. To mitigate randomness, we run each sample 3 times and report the average accuracy.

Table 4: Accuracy on MMLU-Chemistry.

Model	High school	College	Overall
GPT-4o (0-shot)	**88.6**	72.4	80.5
GPT-4o (5-shot)	85.7	72.4	79.0
Claude-3.5-Sonnet (0-shot)	83.8	69.5	76.7
Claude-3.5-Sonnet (5-shot)	83.8	73.3	78.6
ChemCrow (GPT, 0-shot)	47.6	39.0	43.3
ChemCrow (Claude, 0-shot)	69.5	67.6	68.6
ChemAgent (GPT, 0-shot)	80.0	57.1	68.6
ChemAgent (GPT, 5-shot)	87.6	63.8	75.7
ChemAgent (Claude, 0-shot)	73.3	66.7	70.0
ChemAgent (Claude, 5-shot)	86.7	**79.0**	**82.9**

From the results presented in Table 4, we can draw several key observations:

Contrary to expectations, the ChemAgent models frequently underperforms their base LLM counterparts across multiple configurations. Specifically, while ChemAgent achieves the highest overall performance in one specific configuration (Claude, 5-shot), it demonstrates inferior performance compared to the base LLMs in all other configurations. Notably, there exists a substantial performance gap (11.9%) between GPT-4o and the GPT-based ChemAgent in the 0-shot condition. This trend persists across both high school and college subsets, and is also observed with ChemCrow, suggesting a consistent pattern rather than an isolated occurrence. This observation challenges the intuitive assumption that tool augmentation would invariably enhance the capabilities of base LLMs by providing additional valuable information. It also contradicts the expectation that an agent system could default to raw LLM capabilities when tools offer no advantage. Our empirical evidence indicates that this is not uniformly the case, highlighting the requirement of more attention on building tool-augmented agents on certain applications.

Comparing 0-shot and 5-shot performance, the addition of examples (5-shot) yields minimal improvement for base LLMs but results in significant enhancement for ChemAgent. This disparity may be attributed to the extensive pre-training of base LLMs on general chemistry questions, potentially rendering additional examples redundant for task comprehension. Conversely, for ChemAgent, the step-wise demonstration examples appear to effectively guide the LLMs in reasoning and tool utilization, thereby optimizing the problem-solving process. This finding suggests that incorporating examples can be a valuable strategy for enhancing the performance of agent systems.

3.1.3 GPQA-CHEMISTRY

GPQA (Rein et al., 2023) is a challenging dataset that consists of graduate-level questions, requiring advanced knowledge and complex reasoning. We use GPQA-Chemistry, the chemistry questions from the expert-verified GPQA-Diamond subset to evaluate models' abilities in solving difficult chemistry questions. This contains 93 multi-choice questions, ranging from general chemistry to organic and inorganic chemistry. All the evaluated models were prompted to generate CoT solutions and close their output with "the answer is ..." to facilitate answer extraction. We report the average accuracy across 3 runs. The results in Figure 2 can draw some findings:

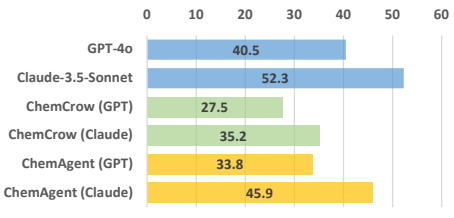


Figure 2: Accuracy on GPQA-Chemistry.

Agent models consistently underperform their base LLM counterparts. This observation holds true for both ChemAgent and ChemCrow, corroborating the results observed in MMLU-Chemistry. These findings suggest that when addressing general chemistry questions, such as those presented in MMLU-Chemistry and GPQA-Chemistry, tool-augmented LLMs may be less effective than unaugmented LLMs. Researchers and practitioners should carefully consider specific application scenarios before implementing tool augmentation for LLMs in this domain.

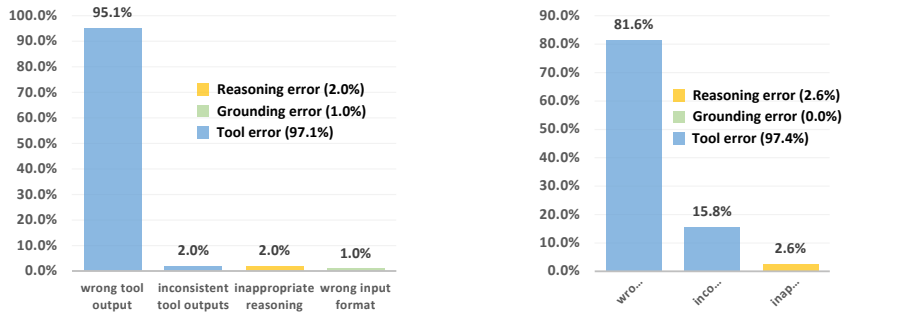
Claude-based models demonstrate consistently superior performance compared to their GPT-based counterparts across both base LLMs and agent configurations. This performance disparity suggests that Claude-3.5-Sonnet may possess more comprehensive chemistry knowledge and exhibit enhanced reasoning capabilities relative to GPT-4o.

3.2 Error Analysis

To take a closer look at how ChemAgent made mistakes and understand more about the reasons, we selected SMolInstruct and MMLU-Chemistry as the representative from each of their categories, and conducted manual error analysis. For each samples where the models failed on, we manually check the error, and based on which module made the error, we classify them into three types. Specifically:

- **Reasoning error**: Errors made by the "reasoning" module, where the agent incorrectly assesses the current situation or formulates an incorrect plan for subsequent steps. For example, misunderstanding the tool output, or proposing an erroneous method.
- **Grounding error**: Errors occurring during tool invocation, such as calling a wrong tool not expected in the "thought", using an incorrect input format, or providing erroneous input to a tool.
- **Tool error**: Errors originating in the environment (the tools in this study), where tools fail to execute or return incorrect/inaccurate information.

3.2.1 SMolInstruct



(a) ChemAgent (GPT) on 102 error cases.

(b) ChemAgent (Claude) on 114 error cases.

Figure 3: The error analysis on SMolInstruct.

We printed out all the samples where ChemAgent made wrong predictions, and manually checked errors for which leaded to the final failure. This involved 102 samples for ChemAgent (GPT) and 114 samples for ChemAgent (Claude), where each sample has 1 error. The result is presented in Figure 3. We can draw the following findings:

We manually analyzed all samples where ChemAgent made incorrect predictions. This analysis encompassed 102 samples for ChemAgent (GPT) and 114 samples for ChemAgent (Claude), with each sample containing one error. The results are presented in Figure 3, from which we can draw the following conclusions:

For both models, tool errors account for over 97% of all errors, highlighting the critical role of tools as essential information sources in these specialized chemistry tasks. This finding underscores the importance of enhancing tool robustness and accuracy. In cases where neural networks serve as tools (e.g., BBBPPredictor, AiExpert) and are inherently subject to imperfect accuracy (as is

prevalent in ChemAgent), it would be beneficial to implement a mechanism that acknowledges potential tool inaccuracies and prompts LLMs to seek alternative methods for information acquisition or verification.

An intriguing observation is the occurrence of errors due to inconsistent outputs from multiple tools, particularly prominent in the Claude-based model. Upon closer examination, this phenomenon is predominantly observed in the Property Prediction-ClinTox (PP-ClinTox) task, where the agent is required to assess molecular toxicity. In these instances, Claude attempted to verify its answer using different tools but encountered information inconsistencies. For example, in some cases, ToxicityPredictor indicated that a molecule was toxic, while WebSearch suggested otherwise, and the LLM chose the incorrect option without employing additional methods for confirmation. the need for improved conflict resolution strategies to better handle inconsistent tool outputs in complex chemistry tasks.

3.2.2 MMLU-CHEMISTRY

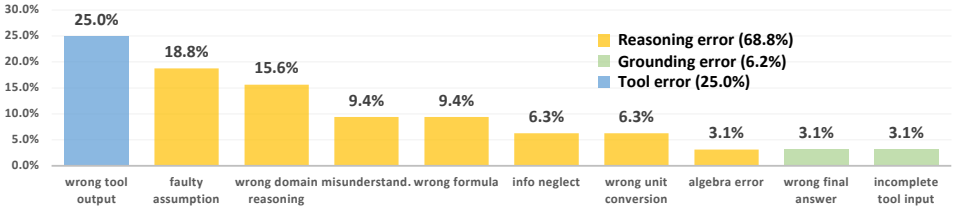


Figure 4: Error analysis of ChemAgent (Claude) on MMLU-Chemistry, calculated on 32 error cases.

To elucidate the error patterns on the general chemistry questions, we conducted a manual analysis of error cases with a domain expert. Our study involves 28 samples where ChemAgent (GPT, 0-shot) failed to provide correct answers. The chemistry expert was invited to meticulously examine the discrepancies between ChemAgent's responses and those of GPT-4o and list the errors. The results of this analysis are presented in Figure 4, from which we can draw several significant observations:

Unlike the SMolInstruct dataset where tool errors predominate, the MMLU-Chemistry dataset reveals a higher proportion of reasoning errors, accounting for nearly 70%. This shift can be attributed to the nature of MMLU tasks, which typically demand broader knowledge and more intricate chemical reasoning while relying less on external tools.

The observed reasoning errors tend to manifest as minor inaccuracies at various intermediate stages of the problem-solving process. Among the 7 reasoning errors in Figure 4, none resulted from an incorrect overall method. Instead, they arose from small mistakes during execution. For instance, "faulty assumptions" occurred when the model applied inapplicable conditions, while "wrong domain reasoning" resulted from incorrect logical reasoning steps. This behavior resembles that of a student who understands the overall concept but makes careless mistakes under exam conditions. Compared to raw LLMs, the tool-augmented agent seems to be more prone to such errors.

Although less prevalent than in the SMolInstruct dataset, tool errors remain a significant portion of the observed errors. This persistence underscores the ongoing need for refinement and enhancement of the tools integrated into the ChemAgent system.

4 RELATED WORK

Recent advancements in large language models (LLMs) have led to the development of sophisticated AI agents capable of assisting in various aspects of chemical research. These agents, such as ChemCrow (M. Bran et al., 2024) and Coscientist (Boiko et al., 2023), have demonstrated the ability to automate routine chemical tasks and accelerate molecular discovery. ChemCrow, for instance, integrates LLMs with common chemical tools to perform a wide range of chemistry-related tasks, consistently outperforming GPT-4 in accuracy. Similarly, Coscientist exemplifies the integration of semi-autonomous robots in planning and executing chemical reactions with minimal human intervention. Other notable agents include Chemist-X (Chen et al., 2024), which focuses on designing

chemical reactions to achieve specific molecules, and ProtAgent (Ghafarollahi & Buehler, 2024), a multi-agent system designed to automate and optimize protein design.

In the realm of experimental planning, several agents have been developed to bridge the gap between virtual assistants and physical laboratory environments. CALMS (CHERUKARA et al., 2024) enhances laboratory efficiency by operating instruments and managing complex experiments through conversational LLMs. BioPlanner (O'Donoghue et al., 2023) improves experimental efficiency by creating pseudocode representations of procedures, while CRISPR-GPT (Huang et al., 2024) assists in designing gene editing experiments iteratively with constant human feedback. LLM-RDF (Ruan et al., 2024) takes this a step further by automating every step of the synthesis workflow, from literature search to product purification.

Cheminformatics tasks have also been significantly impacted by LLM-based agents. CACTUS (McNaughton et al., 2024) automates the application of multiple cheminformatics tools while maintaining human oversight in molecular discovery. ChatMOF (Kang & Kim, 2023) focuses on predicting and generating Metal-Organic Frameworks, integrating MOF databases with its predictor module. IBM ChemChat augments LLMs with common APIs and Python packages used in cheminformatics research, facilitating tasks such as de novo drug design and property prediction. These advancements collectively demonstrate the transformative potential of AI agents in chemical research, streamlining processes, enhancing efficiency, and accelerating scientific discovery.

5 Conclusion

In this paper, we conducted a comprehensive evaluation of tool-augmented language agents for chemistry problem-solving. Our study introduced ChemAgent, an advanced agent leveraging 29 specialized tools, and assessed its performance across diverse chemistry tasks using three datasets: SMolInstruct, MMLU-Chemistry, and GPQA-Chemistry.

Our findings reveal that while ChemAgent substantially outperforms ChemCrow and demonstrates significant improvements on specialized tasks, it does not consistently surpass the base LLMs without tools. The impact of tool augmentation is highly dependent on task characteristics. For tasks requiring specialized molecular operations, tool integration yields notable performance gains. However, for general chemistry questions necessitating extensive reasoning and domain knowledge, tool augmentation may hinder performance.

The error analysis highlights that tool errors predominate in specialized tasks, whereas reasoning errors are more frequent in general chemistry questions. This suggests the need for robust tool implementations and enhanced reasoning capabilities.

Overall, our research underscores the potential and limitations of tool-augmented LLMs in chemistry, emphasizing the importance of task-specific tool selection and integration strategies. Future work should focus on improving tool accuracy and developing mechanisms to balance tool use with intrinsic reasoning abilities to maximize the effectiveness of language agents in chemistry.

REFERENCES

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

Kexin Chen, Junyou Li, Kunyi Wang, Yuyang Du, Jiahui Yu, Jiamin Lu, Lanqing Li, Jiezhong Qiu, Jianzhang Pan, Yi Huang, Qun Fang, Pheng Ann Heng, and Guangyong Chen. Chemist-x: Large language model-empowered agent for reaction condition recommendation in chemical synthesis. *arXiv preprint arXiv:2311.10776*, 2024.

MATTHEW CHERUKARA, ALKATERINI VRIZA, HENRY CHAN, TAO ZHOU, VARUNI KATTI SASTRY, and MICHAEL PRINCE. Calms: Context-aware language model for science. Technical report, Argonne National Laboratory (ANL), Argonne, IL (United States), 2024.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? *arXiv preprint arXiv:2406.04127*, 2024.

Alireza Ghafarollahi and Markus J Buehler. Protagents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

Kaixuan Huang, Yuanhao Qu, Henry Cousins, William A Johnson, Di Yin, Mihir Shah, Denny Zhou, Russ Altman, Mengdi Wang, and Le Cong. Crispr-gpt: An llm agent for automated design of gene-editing experiments. *arXiv preprint arXiv:2404.18021*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Yeonghun Kang and Jihan Kim. Chatmof: An autonomous ai system for predicting and generating metal-organic frameworks. *arXiv preprint arXiv:2308.01423*, 2023.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47:D1102–D1109, 2019.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pp. 1–11, 2024.

Andrew D McNaughton, Gautham Ramalaxmi, Agustin Kruel, Carter R Knutson, Rohith A Varikoti, and Neeraj Kumar. Cactus: Chemistry agent connecting tool-usage to science. *arXiv preprint arXiv:2405.00972*, 2024.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Benedict Emoekabu, Aswanth Krishnan, Mara Wilhelmi, Macjonathan Okereke, Juliane Eberhardt, Amir Mohammad Elahi, Maximilian Greiner, et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Essa Ghareeb, Justin Booth, and Samuel G Rodriques. Bioplanner: automatic evaluation of llms on protocol planning in biology. *arXiv preprint arXiv:2310.10632*, 2023.

Mayk Caldas Ramos, Christopher J Collison, and Andrew D White. A review of large language models and autonomous agents in chemistry. *arXiv preprint arXiv:2407.01603*, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Yixiang Ruan, Chenyin Lu, Ning Xu, Jian Zhang, Jun Xuan, Jianzhang Pan, Qun Fang, Hanyu Gao, Xiaodong Shen, Ning Ye, et al. Accelerated end-to-end chemical synthesis development with large language models. 2024.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

David Weininger. 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, 1988.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *Proceedings of International Conference on Learning Representations*, 2023.

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

A TOOL SET OF CHEMAGENT

The current tool set contains 29 distinct tools, which can be categorized in to general tools, molecule tools, and reaction tools, based on their functions. New tools can be easily added for any applications and tasks.

General tools: Provide broad information retrieval, web searching, and computational.

- AiExpert: A general-purpose AI expert capable of answering a wide range of questions when other specialized tools are insufficient.
- PubchemSearchQA: Searches and retrieves molecule/compound information from PubChem, a comprehensive database of chemical molecules and their activities.
- PythonREPL: Executes Python commands and allows for package installation.
- WebSearch: Searches the internet for both general and domain-specific information, providing concise summaries of relevant content.
- WikipediaSearch: Searches Wikipedia and provides summaries of related content.

Molecule tools: Offer various analyses, predictions, and conversions related to chemical compounds and their properties.

- BBBPPredictor: Predicts the probability of a compound penetrating the blood-brain barrier.
- CanonicalizeSMILES: Converts SMILES representation to its canonical form.
- CompareSMILES: Determines if two molecule SMILES representations are identical.
- CountMolAtoms: Counts the number and types of atoms in a molecule.
- FunctionalGroups: Identifies functional groups present in a molecule.
- GetMoleculePrice: Retrieves the cheapest available price for a purchasable molecule.
- HIVInhibitorPredictor: Predicts the probability of a compound inhibiting HIV replication.
- IUPAC2SMILES: Converts IUPAC names to SMILES representation.
- LogDPredictor: Predicts the octanol/water distribution coefficient (logD) at pH 7.4.
- MolSimilarity: Computes the Tanimoto similarity between two molecules.
- MoleculeCaptioner: Generates a textual description of a molecule using neural networks.
- MoleculeGenerator: Creates SMILES representations based on molecular descriptions using neural networks.
- Name2SMILES: Converts common molecule names to SMILES representation.
- PatentCheck: Verifies if a molecule is patented.
- SELFIES2SMILES: Converts SELFIES representation to SMILES representation.
- SMILES2Formula: Derives the molecular formula from SMILES representation.
- SMILES2IUPAC: Converts SMILES representation to IUPAC name.
- SMILES2SELFIES: Converts SMILES representation to SELFIES representation.
- SMILES2Weight: Calculates the molecular weight from SMILES representation.
- SideEffectPredictor: Predicts the probabilities of a compound causing various side effects across 20 different categories.
- SolubilityPredictor: Predicts the log solubility of a compound in mol/L.
- ToxicityPredictor: Predicts the probability of a compound being toxic.

Reaction tools: Predict products of chemical reactions and suggest potential reactants for synthesizing given products.

- ForwardSynthesis: Predicts the products of a chemical reaction based on given reactants and reagents.
- Retrosynthesis: Conducts single-step retrosynthesis, suggesting potential reactants to synthesize a given product.