ACS OMEGA

Open Access

http://pubs.acs.org/journal/acsodf

Article

CACTUS: Chemistry Agent Connecting Tool Usage to Science

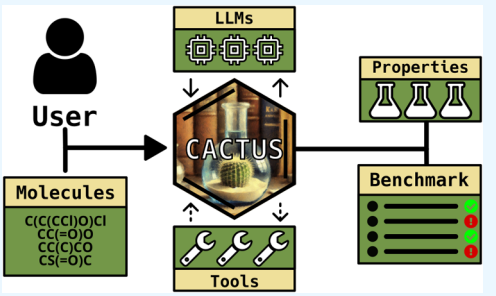
Andrew D. McNaughton, Gautham Krishna Sankar Ramalaxmi, Agustin Kruel, Carter R. Knutson, Rohith A. Varikoti, and Neeraj Kumar*

Cite This: *ACS Omega* 2024, 9, 46563−46573

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Large language models (LLMs) have shown remarkable potential in various domains but often lack the ability to access and reason over domain-specific knowledge and tools. In this article, we introduce Chemistry Agent Connecting Tool-Usage to Science (CACTUS), an LLM-based agent that integrates existing cheminformatics tools to enable accurate and advanced reasoning and problem-solving in chemistry and molecular discovery. We evaluate the performance of CACTUS using a diverse set of open-source LLMs, including Gemma-7b, Falcon-7b, MPT-7b, Llama3-8b, and Mistral-7b, on a benchmark of thousands of chemistry questions. Our results demonstrate that CACTUS significantly outperforms baseline LLMs, with the Gemma-7b, Mistral-7b, and Llama3-8b models achieving the highest accuracy regardless of the prompting strategy used. Moreover, we explore the impact of domain-specific prompting and hardware configurations on model performance, highlighting the importance of prompt engineering and the potential for deploying smaller models on consumer-grade hardware without a significant loss in accuracy. By combining the cognitive capabilities of open-source LLMs with widely used domain-specific tools provided by RDKit, CACTUS can assist researchers in tasks such as molecular property prediction, similarity searching, and drug-likeness assessment.



INTRODUCTION

Large language models (LLMs) are foundation models that are combined under a single paradigm to support various tasks or services. Despite being trained on vast corpora of data, these transformer-based LLMs have a limited understanding of the curated or parsed text.[1] Current research has revealed the possibility of augmenting LLMs with tools that aid in efficiently solving various problems and tasks.[2−4] Previous work has also shown that providing specific prompts, curated toward a specific task, can enhance the time and quality of the text generated by the models.[5] Combining these two approaches is the Tool Augmented Language Model (TALM) framework, detailed in Parisi et al.,[6] which outperforms existing models on the tasks it is configured for. However, with any of these approaches, although the generated answers may appear correct, LLMs fail to reason or demonstrate subject knowledge as is typically demonstrated by humans.[7,8] The robustness failures derived from the statistical associations learned by the model could manifest in a correlated way across several different domains.[9] If foundation models become integrated with important systems that leverage the foundation model's ability to quickly adapt to many different tasks and situations, failures could result in significantly unwanted outcomes.

The resourceful LLMs like GPT4,[10] LLaMA,[11] Gemma,[12] MPT,[13] Falcon,[14] and Mistral[15] show improved performance over a range of activities.[16−18] Despite these strides, the inherent limitations of such models become apparent when faced with challenges that require access to dynamic, real-time, or confidential data, which remain inaccessible within their static training data sets. This gap underscores a critical need for LLMs to evolve beyond their current capacities, leveraging external APIs to fetch or interact with live data, thereby extending their utility in real-world applications.[6] In domain-specific applications, particularly within the chemical, biological, and material sciences, the limitations of LLMs are even more pronounced. The intricate nature of chemical data coupled with the dynamic landscape of drug discovery and development presents a complex challenge that pure computational models alone cannot address effectively. Recognizing this, the integration of cheminformatics tools with the cognitive and analytical abilities of LLMs offers a promising pathway.

At the forefront of this transformation are Intelligent Agents, autonomous entities capable of designing, planning, and executing complex chemistry-related tasks with exceptional efficiency and precision.[19] These systems are not only capable of utilizing a variety of LLMs for specific tasks but also adept at employing APIs and Internet search tools to gather relevant

Received: September 12, 2024
Revised: October 8, 2024
Accepted: October 14, 2024
Published: October 25, 2024

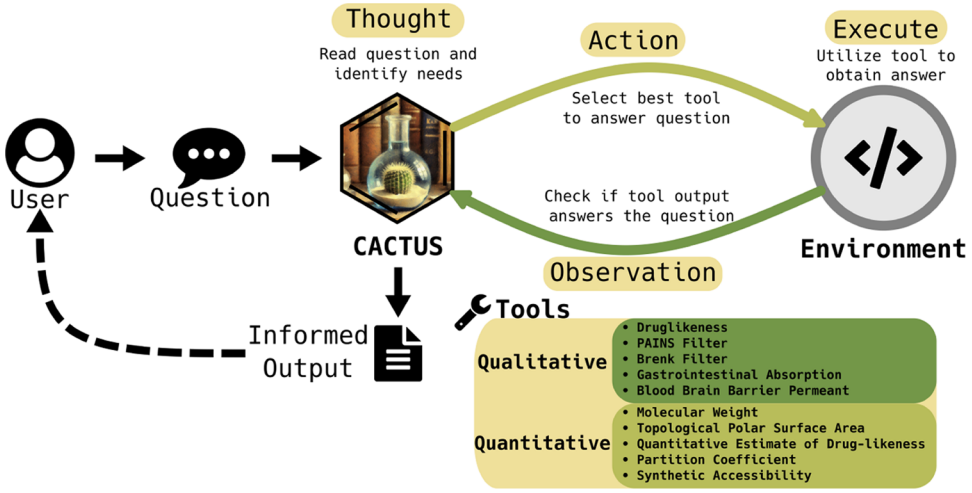


Figure 1. General workflow of the CACTUS agent that details how the LLM interprets input to arrive at the correct tool to use to obtain an answer. Starting from the user input, CACTUS follows a standard "Chain-of-thought" reasoning method with a Planning, Action, Execution, and Observation phase to obtain an informed output.

material and data. For example, integrating an Agent into large, tool-based platforms such as KNIME[20] or Galaxy[21] could form a natural language interface between the user and their analysis. By acting as intermediaries, these Agents could significantly streamline the process of scientific discovery and autonomous experimentation with or without a human in the loop. Toward that end and taking inspiration from ChemCrow,[22] an LLM-assisted chemistry synthesis planner, we have developed an Intelligent Cheminformatics Agent focused on assisting scientists with *de novo* drug design and molecular discovery. Cheminformatics focuses on storing, retrieving, analyzing, and manipulating chemical data. It provides the framework and methodologies to connect computational linguistics with chemical science. This synergistic approach aims to leverage the strengths of both domains by facilitating a more comprehensive and effective exploration of therapeutic compounds, streamlining the drug development process and ultimately accelerating the discovery from conceptualization to clinical application. In this work, we developed Chemistry Agent Connecting Tool Usage to Science (CACTUS), an LLM-powered agent that possesses the ability to intelligently determine the most suitable tools for a given task and the optimal sequence in which they should be applied, effectively optimizing workflows for chemical research and development.

The implications of these intelligent agents are far-reaching. They enable the autonomous operation of complex tasks from data analysis to experimental planning, hypothesis generation, and testing and advance our understanding of what can be achieved through computational chemistry. The synergistic relationship among human intelligence, artificial intelligence, and specialized software tools holds the potential to transform the landscape of drug discovery, catalysis, material science, and beyond. This relationship and combination of domains make the molecular discovery process more efficient, accurate, and innovative. As we stand on the precipice of this new era in cheminformatics, the integration of LLMs and computational tools through intelligent agents such as CACTUS promises to unlock a future where the limits of scientific discovery are bound only by the depths of our imagination.

METHODS

Tool-augmented language models consist of two major components: external tools and language models. This section will discuss the approaches used to implement the language model agent and provide a focused look at the tools used. We also go into great detail about the strategies used when prompting our agent and how we performed benchmarking. Each of these steps is a critical component of forming a complete intelligent agent able to solve a wide range of problems with the added ability of quick model swapping.

Agent. An important consideration when building a TALM is the framework in which it will be implemented. We have selected the commonly used open-source platform, LangChain,[23] for this purpose. This framework simplifies the integration of prompts with LLMs through a comprehensive set of prebuilt Python modules known as "chains." It also provides convenient integration with popular LLM hosting/inference platforms such as the OpenAI API and HuggingFace Transformers.[24] CACTUS utilizes LangChain's implementation of a custom MRKL agent,[25] which can be broken into 3 parts: tools, LLMChain, and agent class. The tools in this instance are a collection of cheminformatics helper functions that wrap well-known Python libraries into well-described tools for agents to use. These tools are explained in much more detail in Section 2.2. The LLMChain is a LangChain-specific feature that combines the base LLM and a prompt template to form one unit. It is used to instantiate the model and parse user input when any inference. In CACTUS, we provide a prompt that guides the agent to answer cheminformatics questions by describing the typical steps involved in answering such questions. The last requirement for CACTUS is the agent class. These are also LangChain-implemented functions that are used to interpret user input after the initial prompt and make decisions on which actions to take to best solve the question. CACTUS sticks with a general-purpose implementation of the zero-shot agent class that uses the ReAct[26] framework to determine which tool to use from the tool's description. This combination of tools, LLMChain, and a zero-shot agent makes CACTUS an extensible LLM tool that can quickly integrate new tools to solve a range of cheminformatics questions. The generalized workflow can be seen in Figure 1.

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573

Here, we introduce a mathematical formulation to describe the key components and processes of the CACTUS framework:

Let us consider $\mathcal{T} = t_1, t_2, ..., t_n$ as the set of cheminformatics tools available to CACTUS as discussed above, where each tool t_j is a function that takes a tool-specific input x_i and produces an output y_i:

$$t_j: x_i \rightarrow y_i \quad (1)$$

The LLMChain is represented as a function L that takes a user-specificed input u and a set of tools $\mathcal{T}$ as input and outputs a sequence of actions $\mathcal{A} = a_1, a_2, ..., a_m$:

$$L(u, \mathcal{T}) = \mathcal{A} \quad (2)$$

Note: In our framework, LLMChain represents the combination of the LLM with a prompt, so we use L to represent LLMChain, which is equivalent to the LLM in this context.

Each action a_i in the sequence $\mathcal{A}$ corresponds to the application of a specific tool t_j on the tool input x_i, resulting in an output y_i,

$$a_i: (t_j, x_i) \rightarrow y_i \quad (3)$$

The zero-shot agent class is modeled as a function Z that takes the user input u, the set of tools $\mathcal{T}$, and the set of actions $\mathcal{A}$ as input to produce the final output o,

$$Z(u, \mathcal{T}, \mathcal{A}) = o \quad (4)$$

The final output o is the result of executing the sequence of actions $\mathcal{A}$ determined by the LLMChain, given the user input u and the available tools $\mathcal{T}$.

The ReAct framework used by the zero-shot agent class was represented as a function R that takes the user input u, the set of tools $\mathcal{T}$, and the tool descriptions $\mathcal{D} = d_1, d_2, ..., d_n$ as input and outputs the most appropriate tool t_k to use,

$$R(u, \mathcal{T}, \mathcal{D}) = t_k \quad (5)$$

This combination of cheminformatics tools, LLMChain, and a zero-shot agent makes CACTUS an extensible LLM tool that can quickly integrate new tools to solve a range of cheminformatics questions.

Cheminformatics Tools. CACTUS is designed to empower chemistry and cheminformatics researchers by seamlessly integrating familiar tools from widely used libraries like RDKit[27] into an intuitive chat-based interface. We prioritize open-source Python tools to make CACTUS accessible and adaptable. Our focus is not on improving the accuracy of these existing tools but rather on demonstrating how an LLM agent can intelligently leverage them within a more streamlined workflow. For detailed accuracy assessments of the individual tools, please refer to the original publications provided in Table 1 and RDKit documentation.

The tool set provided to the CACTUS agent consists of ten different tools providing information on various descriptors for any given chemical compound used as input. Table 1 contains the list of currently available tools that can assist in obtaining different physiochemical properties and molecular descriptors of the input chemical compounds. This includes molecular weight, log of the partition coefficient (LogP), topological polar surface area (TPSA), quantitative estimate of drug-likeness (QED), and synthetic accessibility (SA) of the input chemical compounds. Moreover, using the BOILED-Egg method, CACTUS can also estimate the pharmacokinetic properties like blood−brain barrier permeability and gastrointestinal absorption of any given chemical compound.[28] Our model also implements drug-likeness, PAINS, and Brenk filters to identify structural and toxicity alerts. All of these tools in our model assist in identifying and screening both currently available and new lead compounds. Currently restricted to using a simple SMILES as input, future releases will allow for varied user input (compound name, molecular formula, InChI key, CAS number, SMILES, ChEMBL ID, or ZINC ID) where the agent will first convert it to SMILES notation and then used as input for the available tools. While these tools leverage existing code snippets from RDKit, their accuracy is limited to the underlying methods. However, CACTUS's flexible design allows for seamless integration of new tools as more accurate methods become available.

Table 1. Cheminformatics Tools Currently Supported by CACTUS. These Tools Provide a Comprehensive Assessment of Molecular and Physicochemical Properties[a]

tool name	description
MolWt[27]	float [0, ∞]—molecular weight
LogP[29]	float [−∞, ∞]—predicted partition coefficient
TPSA[30]	float [0, ∞]—topological polar surface area
QED[29,31]	float [0, 1]—quantitative estimate of drug-likeness
SA[32]	float [1, 10]—synthetic accessibility
BBB permeant[28]	string [yes, no]—is in "yolk" of BOILED-egg model
GI absorption[28]	string [low, high]—is in "white" of BOILED-egg model
drug-likeness[33]	boolean—passes Lipinski rule of 5
Brenk filter[34]	boolean—passes brenk filter
PAINS filter[35]	boolean—passes PAINS filter

[a]All tools require input in the SMILES format. By leveraging these tools, CACTUS enables researchers to make informed decisions in the molecular discovery process and prioritize compounds with the most promising characteristics. Accuracy of these methods is outlined in the RDKit documentation as well as the citations provided in the tool name column.

Prompting Strategy. One important aspect investigated was the significance of the prompt for the agent. Through the LangChain implementation of LLM agents, there is a default prompt that provides a generic instruction about what tools are available and what the task of LLM is. However, this is not necessarily required for understanding domain-specific information. To test the hypothesis, we ran 2 scenarios: one where we left the default prompt unchanged and only included tool descriptions (Minimal Prompt), and one where we modified the prompt to align the agent more with the domain of chemistry (Domain Prompt). The belief is that a domain-aligned prompt will steer the LLM toward a better interpretation of the questions being asked and therefore be more effective in answering user queries. Since we were using a wide range of LLMs for testing, the minimal prompt also included model-specific tokens so that we were not unfairly evaluating models against the domain prompt. The current focus of the paper is a natural language interface to cheminformatics tool sets, and the future iterations of CACTUS for inverse design tasks have been discussed.

Benchmarking. Evaluating domain-specific TALMs can be challenging, but we can follow the examples set by general benchmarking suites.[3,36−38] To achieve this, we created sets of questions that mimic typical queries the agent would encounter and measure how many it could answer correctly

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573

without needing extra prompting from the user (i.e., having to rephrase the typed question to get a correct answer).

For CACTUS, we generated three sets of cheminformatics questions that test the ability of the agent to answer domain-specific queries. The first set contains qualitative questions that require answers like Yes/No or True/False. The second set includes quantitative questions that require the agent to interpret numerical values. The third set is a concatenation of both the qualitative and quantitative sets, which we call the combined set.

We used the qualitative and quantitative sets separately to evaluate how the model performs with tools specific to each type of question. This means the agent will not have access to quantitative tools when answering qualitative questions and vice versa. In the combined data set, however, all tools are available for all questions, providing a comprehensive assessment.

Table 2 highlights examples of questions passed as user input into the CACTUS agent. The qualitative and quantitative data sets each contain 500 questions like the ones shown, and the combined data set contains the combined 1000 from the previous two data sets. Most tests were done on the combined data set as we want to test the LLM agent's ability to perform a diverse set of tasks.

Table 2. Table Demonstrating Examples of the Questions Asked of the CACTUS Agent in the Cheminformatics Benchmark Used in This Paper

qualitative questions		
question	step	answer
does CCON=O pass the blood−brain barrier?	use BBB tool w/SMILES	yes
what is the GI absorption of C#C?	use GI tool w/SMILES	low
quantitative questions		
question	step	answer
what is the QED of CCCC=O?	use QED tool w/SMILES	0.44
what is the TPSA of C(CS)O	use TPSA tool w/SMILES	20.23

To construct these data sets, we take a set of 1000 compounds from PubChem[39] and randomly sample 100. We then populate question templates with the SMILES string representation of the compound. This allows us to generate 100 questions for each type of tool listed in Table 1, resulting in 1000 questions for the combined data set. To obtain the answers to these questions, we pass the same set of SMILES data through a script that simply uses the Python-wrapped tool directly to obtain the expected output. These answers are not externally validated as this work is not focused on the detailed benchmarking of the open-source tools used to calculate these properties but on the ability of the agent to utilize the tool as intended to come up with the expected answer. These benchmarks are able to be programmatically generated, so any list smiles can be used to create new data sets.

Tool-Specific Benchmarking. While comprehensive tool-specific accuracy benchmarking is beyond the scope of this work, we have chosen RDKit for its extensive use and established reliability within the cheminformatics community. Numerous studies have leveraged RDKit for tasks similar to those addressed by our agent either directly or for training newer models.[40−44] For detailed performance evaluations of RDKit's methods, we direct readers to the official documentation and original publications.

RESULTS AND DISCUSSION

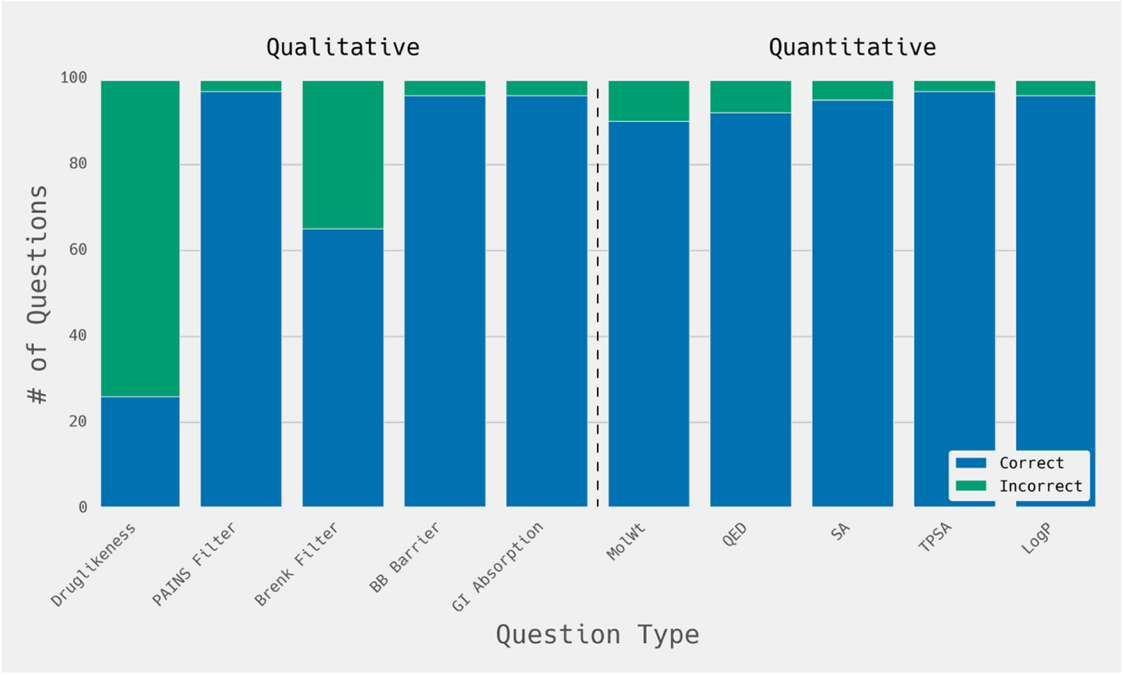
The implementation of CACTUS represents a significant step forward in the field of cheminformatics, offering a powerful and flexible tool for researchers and chemists engaged in molecular discovery and drug design. The benchmarking studies conducted on various 7 billion parameter models demonstrate the robustness and efficiency of the CACTUS framework, highlighting its potential to streamline and accelerate the drug discovery process as an example.

Benchmarking and Performance Evaluation. The performance of CACTUS was evaluated using a comprehensive set of 1000 questions, covering 10 different tools (Table 1, with and without a domain-specific prompt on each model, shown in Figure 2). Correct answers were scored as correct, while wrong answers, inability to converge on an answer, or inability to use the provided tool correctly were marked as incorrect. In this article, we did not differentiate between incorrect tool usage and simply providing a wrong answer. Any answers that did not coherently address the question were considered incorrect. We accepted correct answers that contained additional formatted text after the correct answer although this is not the preferred format. This additional information can be programmatically removed before returning the response to the user, or further prompts can be engineered to reduce additional text. Each type of question in the full question set was asked 100 times, resulting in 10 types of questions corresponding to the 10 tools provided in Table 1. This approach allowed us to identify which tools posed a greater challenge for the model and where improvements to either the tool description or model prompt could be made.

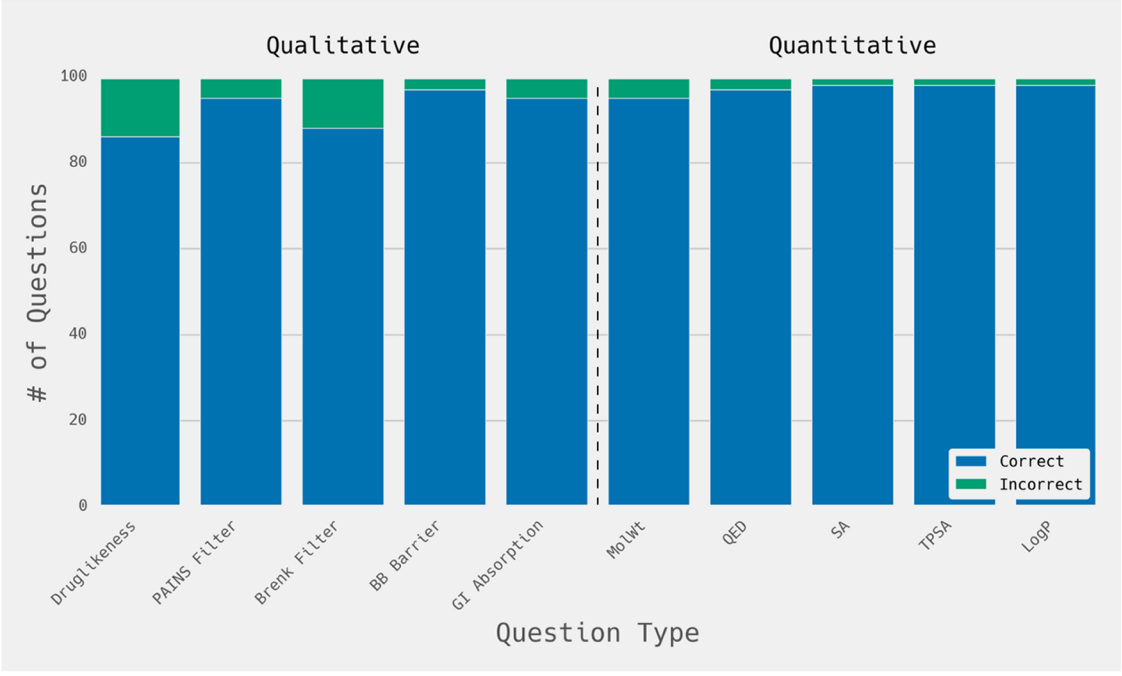
The results shown in Figure 2 highlight the importance of domain-specific prompting in improving the accuracy of the model's responses, particularly for qualitative questions. This finding aligns with recent research emphasizing the role of prompt engineering in enhancing the performance of language models.[45]

In the progression of AI and its applications in scientific inquiry, it is crucial to analyze the comparative effectiveness of various models in handling domain-specific tasks. The benchmarking analysis presented in Figure 3 offers significant insights into the performance of different language models when prompted with both minimal and domain-specific information. A comprehensive review of the performance data across the full spectrum of question types reveals that the Gemma-7b, Mistral-7b, and Llama3-8b models showcase robustness and versatility, performing admirably regardless of the nature of the prompt. Their consistent accuracy across different types of questions ranging from physiochemical properties like drug-likeness and blood−brain barrier permeability to more complex metrics like a quantitative estimate of drug-likeness (QED) highlight their reliability for a broad range of inquiries within the domain of molecular science. In contrast, models such as Falcon-7b exhibit a noticeable disparity between performances with minimal and domain prompts. This variability suggests that Falcon-7b, while capable, may require more fine-tuned prompting to leverage its full potential effectively. The substantial difference in performance based on the prompt type points to an intrinsic model sensitivity to input structure and content, which can be pivotal in crafting effective inquiry strategies. Furthermore, the successful deployment of smaller models, such as Phi2 and OLMo-1b, on consumer-grade hardware (Figure 4) highlights

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573



(a) Benchmark performance on the Gemma-7b model with a minimal prompt on each of the 10 question types.



(b) Benchmark performance on the Gemma-7b model with a domain prompt on each of the 10 question types.

Figure 2. Comparison of the Gemma-7b model with different prompting strategies on the full question set benchmark shows significant improvement in the qualitative question set when comparing the minimal prompt (a) to the domain prompt (b), while demonstrating a similar performance in the quantitative question set.

the potential for democratizing access to powerful chem-informatics tools, enabling researchers with limited computational resources to harness the capabilities of CACTUS.

Open Source Models in Varied Settings. This comprehensive model comparison and analysis have broader implications for the employment of open-source models in scientific environments. The ability of models to perform well with domain-specific prompts is particularly encouraging, as it implies that with proper configuration, open-source models can be highly effective tools. The adaptability demonstrated by the Gemma-7b, Mistral-7b, and Llama3-8b models indicates their potential for widespread applicability across various

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573

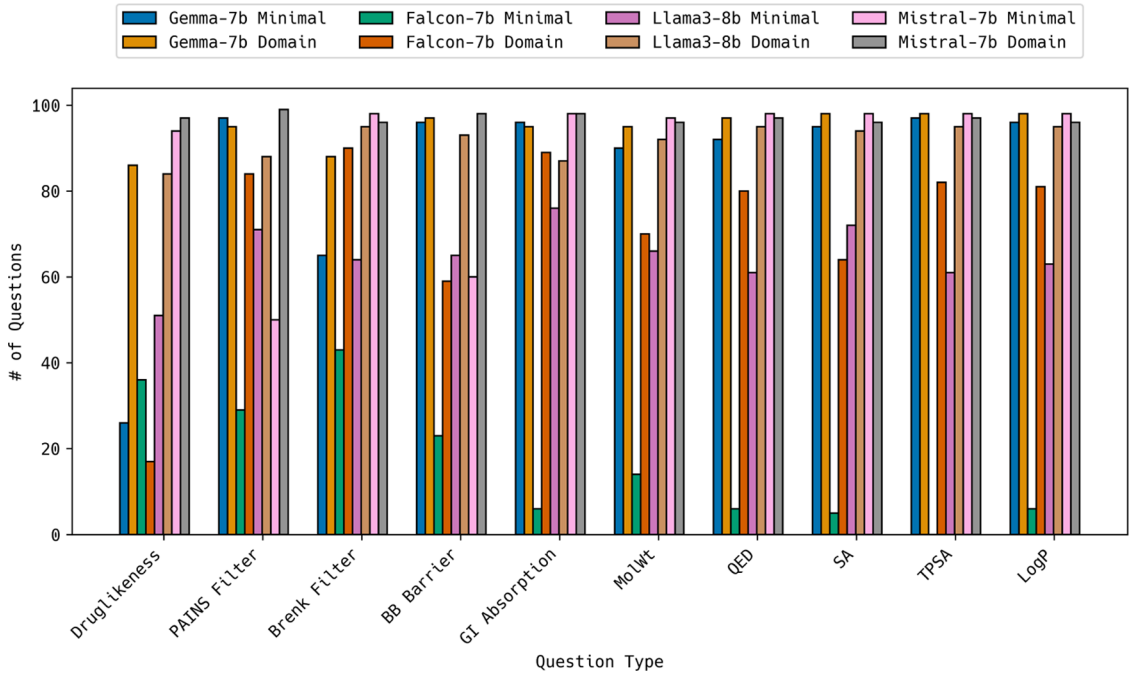


Figure 3. Comparison of model performance among 7b parameter models using minimal and domain-specific prompts. The Gemma-7b, Mistral-7b, and Llama3-8b models demonstrate strong performance and adaptability across prompting strategies, highlighting their potential for widespread applicability in various computational settings, from high-performance clusters to more modest research setups.

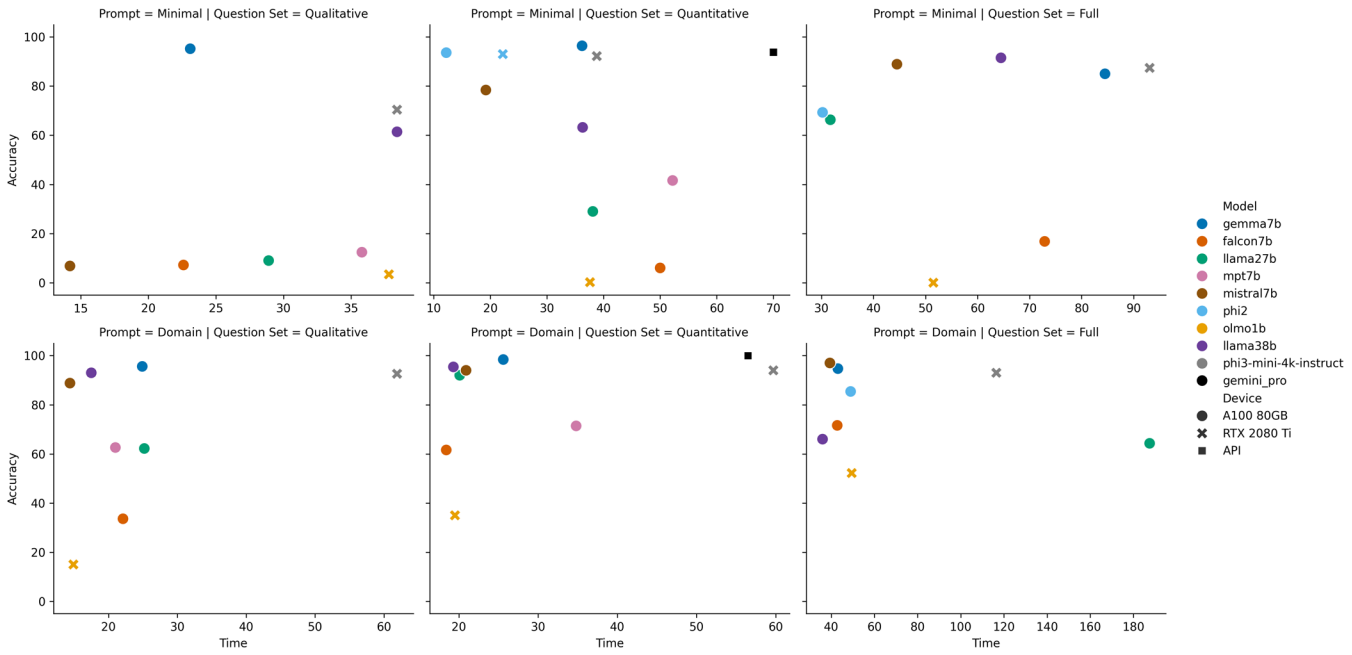


Figure 4. Comparison of the model performance using the accuracy and execution time as key metrics. The study evaluates various open-source models available on the HuggingFace including Gemma-7b, Falcon-7b, MPT-7b, Llama3-8b, and Mistral-7b, phi2, and olmo1b. Different combinations of conditions, such as model type (Vicuna, LLaMa, MPT), prompting strategy (minimal or domain-specific), GPU hardware (A100 80GB or RTX 2080 Ti), and benchmark size (small or large), were used to assess the model's capabilities.

computational settings from high-performance clusters to more modest research setups. Moreover, the ability to effectively prompt open-source models opens the door to their use in a variety of scientific contexts. It allows researchers to customize models to their specific domain, potentially bridging the gap between generalized AI capabilities and specialized knowledge areas.

The flexibility and performance of these models have significant implications for scientific research, particularly in fields such as synthetic organic chemistry and drug discovery. For researchers in these domains, the ability to use open-source models effectively can accelerate the discovery process, enhance predictive accuracy, and optimize computational resources. The insights from this benchmarking study provide a roadmap for selecting and tailoring models to specific

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573

research needs, thereby maximizing their utility in advancing scientific goals. The benchmarking study of the selected 7b parameter models serves as a testament to the progress in AI-driven research tools. It highlights the necessity of prompt optimization and the promise of open-source models in diverse scientific inquiries. The analysis underscores the potential of these models to become integral components in the computational chemist's toolkit, paving the way for innovative breakthroughs in molecular design and drug discovery.

Hardware Performance and Model Efficacy. The deployment of CACTUS models through vLLM offers a significant advantage by optimizing the performance across a variety of GPUs used for LLM inference. In our benchmarking studies, we utilized two types of NVIDIA GPUs: the data center-grade A100 80GB and the consumer-grade RTX 2080 Ti. Our objective was to evaluate the performance of models under different combinations of model size, GPU type, and prompting strategy (minimal or domain-specific). The performance metric was determined by the inference speed in relation to the model's accuracy. Figure 4 shows the summary of LLMs deployed under different conditions (GPU hardware used, prompt, and benchmark set used) and how well they performed. The efficiency of these models across diverse hardware platforms highlights their potential for widespread implementation in a range of research settings.

The models evaluated include Gemma-7b, Falcon-7b, MPT-7b, Llama2-7b, Llama3-8b, and Mistral-7b, as well as three smaller models, Phi2, Phi3, and OLMo-1b. The inclusion of these smaller models highlights the potential for successfully deploying models on local resources with limited computational power (e.g., consumer-grade GPUs such as the RTX 2080 Ti) while still achieving accurate results. We also include a comparison to the API-based Gemini-Pro model for a small comparison of a proprietary model to the selected open-source models. Overall, the model performance was found to be relatively quick on both the 500-question sets (Qualitative/Quantitative) and the 1000-question combined set (Full), with the occasional model taking a comparatively longer amount of time to finish. A full list of the data used to plot these summary figures can be found in Supporting Information.

The most interesting outcome is that smaller models deployed on consumer-grade hardware (RTX 2080 Ti) do not perform drastically worse than their larger parameter model counterparts deployed on the a100 80GB hardware. Looking at the performance of the Phi2 (2.7B parameters) and Phi3 (3.8B parameters) models, it quickly and accurately tackles the 500 question quantitative benchmark with similar performance regardless of the GPU used with the a100 80GB version, unsurprisingly as the fastest. Another interesting outcome is the performance of the OLMo-1b parameter model on the combined question set and the RTX 2080 Ti GPU. While unable to obtain any correct answers for the minimal prompt, it jumps up to a surprising 52.2% accuracy when a domain-specific prompt is used. These results indicate that these smaller models can be deployed locally by users and still be able to interpret questions, possibly by providing more specialized prompts. The Gemini-Pro API model was very accurate for both the minimal and domain prompt but was impacted by quota restrictions, causing a dramatic increase in time for the quantitative benchmark. This time of inference limitation could be seen as a strength of hosting open-source models locally, where the user is only restricted by hardware.

In general, inference time increased as the question set size increased (e.g., from quantitative/qualitative to full), while accuracy tended to decrease with longer inference times. Domain prompts achieved faster inference and accuracy than minimal prompts for models such as Falcon-7b, MPT-7b, and Mistral-7b. However, there was an exception in the case of the Phi2 model on the full question set, where the minimal prompt resulted in a faster inference but lower accuracy.

The hardware performance analysis highlights the importance of considering the interplay between model size, GPU capabilities, and prompting strategies when deploying CACTUS models for molecular property prediction and drug discovery. The ability to achieve accurate results with smaller models on consumer-grade hardware opens up the possibility of wider adoption and accessibility of CACTUS for researchers with limited computational resources. Furthermore, the impact of domain-specific prompting on both inference speed and accuracy emphasizes the need for carefully designed prompts tailored to the specific application domain. As CACTUS continues to evolve and integrate with other computational tools and autonomous discovery platforms, optimizing hardware performance will remain a critical consideration. Future research should explore the development of more efficient algorithms and architectures (energy efficiency) for deploying CACTUS models on a variety of hardware configurations, ensuring that the benefits of this powerful tool can be realized across a wide range of research settings and computational resources.

Issues Encountered and Resolutions. During the development and benchmarking of the CACTUS agent using open-source models and the LangChain framework, several key challenges were identified. These issues, along with the solutions implemented, provide valuable insights for researchers and developers working on similar workflows.

One of the primary issues encountered was the slow inference speed when hosting open-source language models locally on machines utilizing CPUs. Most APIs quickly provide inference results when making calls, and this is not something locally hosted models typically replicate well, especially when running on CPUs over GPUs. For this work, we initially used models from HuggingFace and deployed them through the HuggingFace Pipelines Python package. This allowed us to serve models, but the inference time was quite slow when the samples were wrapped in the LangChain agent. To address this, we began utilizing vLLM to host HuggingFace models instead. This substantially decreased our inference time and allowed for API-like response times from models, even those hosted on less powerful consumer-grade GPU hardware.

The second major challenge was related to prompt engineering. Our results shown previously highlight that for some models the prompt has a great effect on not only the model accuracy but also the inference time. We spent a good amount of time trying to hone our prompting strategy to yield consistently accurate and efficient results with mixed effects. We ended up needing specialized prompts for each open-source LLM we used, as some were fine-tuned much differently than others and required a very specific prompt style to return usable results.

These challenges highlight the need for continued research and development in the areas of model deployment and prompt engineering. Future work will be focused on optimizing the deployment of open-source models on various hardware configurations, including CPUs and GPUs, to ensure

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573

that CACTUS can be efficiently utilized across a wide range of computational resources. This may involve the development of novel algorithms and architectures that can better leverage the capabilities of different hardware setups as well as the creation of more user-friendly tools and frameworks for model deployment and management. In terms of prompt engineering, the development of standardized prompt templates and best practices for prompt engineering in the context of molecular property prediction and drug discovery could help streamline the development process and improve the consistency of results across different models and data sets.

Future Outlook—Molecular Design. CACTUS has already demonstrated its potential in estimating basic metrics for input chemical compounds, but its future lies in its evolution into a comprehensive, open-source tool specifically designed for chemists and researchers working on therapeutic drug design and discovery. This will be achieved by the integration of physics-based molecular AI/ML models, such as three-dimensional 3D-scaffold,[46] reinforcement learning,[47] and graph neural networks (GNNs)[48] accompanied by molecular dynamics simulations, quantum chemistry calculations, and high-throughput virtual screening.[48−52] Such capabilities are essential for accurately modeling molecular interactions and predicting the efficacy and safety of potential therapeutic agents.[53]

The development plan also includes implementing advanced functionalities for identifying compounds that exhibit structural and chemical similarities as well as pinpointing key fragments crucial for biological activity. This feature will allow researchers to explore a vast chemical space more efficiently, identifying lead compounds with higher precision. These additions are expected to significantly accelerate and deepen the agent's ability to understand compound behaviors in 3D spaces and allow researchers to develop more comprehensive and effective workflows for drug discovery and materials design. Additionally, we plan to include tools that identify key fragments and compounds with similar structural and chemical features from the vast available chemical databases. Tools that can calculate physiochemical and pharmacokinetic properties and about 60 other descriptors will be added to the agent to identify quantitative structure−activity relationship (QSAR) and quantitative structure−property relationship (QSPR) to help us with screening the compounds and identifying toxic groups.

Beyond these technical enhancements, there is a focus on making CACTUS more explainable and capable of symbolic reasoning. The aim is to address common criticisms of LLMs, particularly their struggle with reasoning and providing explainable outputs. By integrating more advanced symbolic reasoning capabilities, CACTUS will not only become more powerful in its predictive and analytical functions but also provide users with understandable and logical explanations for its recommendations and predictions. This feature would automate the process of predicting how small molecules, such as drug candidates, and interact with targets such as proteins, thereby providing invaluable insights into the potential efficacy of new compounds.

The applications of CACTUS extend beyond drug discovery and can be leveraged in other domains, such as chemistry, catalysis, and materials science. In the field of catalysis, CACTUS could aid in the discovery and optimization of novel catalysts by predicting their properties and performance based on their structural and chemical features.[54] Similarly, in materials science, CACTUS could assist in the design of new materials with desired properties by exploring the vast chemical space and identifying promising candidates for further experimental validation.[55]

The future development of CACTUS is geared toward creating an intelligent, comprehensive cheminformatics tool for molecular discovery that not only aids in the identification and design of therapeutic drugs but also ensures a high degree of safety and efficacy. Through the integration of advanced computational techniques and models, alongside improvements in usability and explainability, CACTUS is set to become an indispensable resource in the quest for novel, effective, and safe therapeutic agents as well as in the discovery and optimization of catalysts and materials.

CONCLUSIONS

In this article, we have introduced CACTUS, an innovative open-source agent that leverages the power of large language models and cheminformatics tools to revolutionize the field of drug discovery and molecular property prediction. By integrating a wide range of computational tools and models, CACTUS provides a comprehensive and user-friendly platform for researchers and chemists to explore the vast chemical space for molecular discovery and identify promising compounds for therapeutic applications.

We assessed CACTUS performance using various open-source LLMs, including Gemma-7b, Falcon-7b, MPT-7b, Llama2-7b, and Mistral-7b, across a set of 1000 chemistry questions. Our findings indicate that CACTUS outperforms baseline LLMs significantly, with the Gemma-7b and Mistral-7b models achieving the highest accuracy regardless of the prompting strategy employed. Additionally, we investigated the impact of domain-specific prompting and hardware configurations on model performance, highlighting the importance of prompt engineering and the potential for deploying smaller models on consumer-grade hardware without a significant loss in accuracy. The ability to achieve accurate results with smaller models such as Phi on consumer-grade hardware opens up the possibility of wider adoption and accessibility of CACTUS, even for researchers with limited computational resources.

One of the key takeaways from the development and benchmarking of CACTUS is the importance of addressing the challenges associated with model deployment and prompt engineering. The solutions implemented in this work, such as the use of vLLM for hosting models and the development of tailored prompts for each open-source LLM, serve as a valuable foundation for future efforts in this field. As the field of AI continues to evolve rapidly, it is essential to keep abreast of new developments in language modeling and related technologies to further enhance the capabilities and performance of CACTUS. The development and benchmarking of CACTUS also highlight key challenges in integrating open-source LLMs with domain-specific tools, such as optimizing inference speed and developing effective prompting strategies. We discussed the solutions implemented to address these challenges, including the use of vLLM for model hosting and the creation of tailored prompts for each LLM.

Looking ahead, the future of CACTUS is incredibly promising, with the potential to transform not only drug discovery but also various other domains, such as chemistry, catalysis, and materials science. The integration of advanced physics-based AI/ML models, such as 3D-scaffold, reinforcement learning, and graph neural networks, will enable a deeper

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573

understanding of compound behaviors in 3D spaces, leading to more accurate predictions of molecular interactions and the efficacy and safety of potential therapeutic agents. Moreover, the addition of tools for identifying key fragments, calculating molecular properties, and screening compounds for toxic groups will significantly enhance the efficiency and precision of the drug discovery process. The focus on improving the explainability and symbolic reasoning capabilities of CACTUS will address common criticisms of large language models and provide users with understandable, logical explanations for the tool's recommendations and predictions.

As CACTUS continues to evolve and integrate with other computational tools and autonomous discovery platforms, it has the potential to revolutionize the way we approach drug discovery, catalyst design, and materials science. By leveraging the power of AI and machine learning, CACTUS can help researchers navigate the vast parameter spaces associated with complex chemical systems, identifying promising candidates for experimental validation and optimization. The future development of CACTUS is geared toward creating an intelligent, comprehensive cheminformatics tool that ensures a high degree of safety and efficacy in the identification and design of therapeutic drugs, catalysts, and materials for various applications. Through the integration of advanced computational techniques and models, alongside improvements in usability and explainability, CACTUS is set to become an indispensable resource for researchers across various scientific disciplines.

In summary, CACTUS represents a significant milestone in the field of cheminformatics, offering a powerful and adaptable tool for researchers engaged in drug discovery, molecular property prediction, and beyond. As we continue to advance AI-driven scientific discovery, agents like CACTUS will play a pivotal role in shaping the future of research, innovation, and human health. By embracing the potential of open-source language models and cheminformatics tools, we can accelerate the pace of scientific advancement and unlock new frontiers in the quest for novel, effective, and safe therapeutic agents, catalysts, and materials.

■ ASSOCIATED CONTENT

Data Availability Statement

The code to run CACTUS and the associated benchmark data can be found on GitHub: https://github.com/pnnl/cactus.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.4c08408.

Comparisons of performance before and after prompting (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Neeraj Kumar − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States;* orcid.org/0000-0001-6713-2129; Email: neeraj.kumar@pnnl.gov

Authors

Andrew D. McNaughton − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States;* orcid.org/0000-0002-4146-7921

Gautham Krishna Sankar Ramalaxmi − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States*

Agustin Kruel − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States;* orcid.org/0000-0002-5571-7418

Carter R. Knutson − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States;* orcid.org/0000-0002-1953-2272

Rohith A. Varikoti − *Pacific Northwest National Laboratory, Richland, Washington 99354, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.4c08408

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This research was supported by the I3T Investment, under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). The computational work was performed using PNNL's research computing at Pacific Northwest National Laboratory. The initial concept of integrating LLM and tools received support from the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative. PNNL is a multiprogram national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

■ REFERENCES

(1) Chiesurin, S.; Dimakopoulos, D.; Sobrevilla Cabezudo, M. A.; Eshghi, A.; Papaioannou, I.; Rieser, V.; Konstas, I. In *The Dangers of Trusting Stochastic Parrots: Faithfulness and Trust in Open-domain Conversational Question Answering*, Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, 2023; pp 947−959.

(2) Mialon, G.; Dessi, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Roziere, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; Grave, E.; LeCun, Y.; Scialom, T. Augmented Language Models: a Survey. In *Transactions on Machine Learning Research*; Survey Certification, 2023.

(3) Xu, Q.; Hong, F.; Li, B.; Hu, C.; Chen, Z.; Zhang, J. On the Tool Manipulation Capability of Open-Source Large Language Models. 2023, arXiv:2305.16504. arXiv.org e-Print archive. http://arxiv.org/abs/2305.16504.

(4) Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B. et al. Toolllm: Facilitating Large Language Models to Master 16000. Real-World Apis. 2023, arXiv:2307.16789. arXiv.org e-Print archive. http://arxiv.org/abs/2307.16789.

(5) Cai, T.; Wang, X.; Ma, T.; Chen, X.; Zhou, D. Large Language Models as Tool Makers. 2023, arXiv:2305.17126. arXiv.org e-Print archive. http://arxiv.org/abs/2305.17126.

(6) Parisi, A.; Zhao, Y.; Fiedel, N. Talm: Tool Augmented Language Models. 2022, arXiv:2205.12255. arXiv.org e-Print archive. http://arxiv.org/abs/2205.12255.

(7) Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.; Song, X.; Zhou, D. Large Language Models Cannot Self-Correct Reasoning

Yet. 2023, arXiv:2310.01798. arXiv.org e-Print archive. http://arxiv.org/abs/2310.01798.
(8) Kambhampati, S. Can large language models reason and plan? *Ann. N.Y. Acad. Sci.* **2024**, *1534*, 15−18.
(9) Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; Bohg, J.; Bosselut, A.; Brunskill, E. et al. On the Opportunities and Risks of Foundation Models. 2021, arXiv:2108.07258. arXiv.org e-Print archive. http://arxiv.org/abs/2108.07258.
(10) Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S. et al. Gpt-4 Technical Report. 2023, arXiv:2303.08774. arXiv.org e-Print archive. http://arxiv.org/abs/2303.08774.
(11) Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F. et al. Llama: Open and Efficient Foundation Language Models. 2023, arXiv:2302.13971. arXiv.org e-Print archive. http://arxiv.org/abs/2302.13971.
(12) Team, G.; Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J. et al. Gemma: Open Models Based on Gemini Research and Technology. 2024, arXiv:2403.08295. arXiv.org e-Print archive. http://arxiv.org/abs/2403.08295.
(13) Team, M. N. Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs, 2023. www.mosaicml.com/blog/mpt-7b.
(14) Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Étienne Goffinet; Hesslow, D.; Launay, J.; Malartic, Q.; Mazzotta, D.; Noune, B.; Pannier, B.; Penedo, G. The Falcon Series of Open Language Models. 2023, arXiv:2311.16867. arXiv.org e-Print archive. http://arxiv.org/abs/2311.16867.
(15) Jiang, A.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L. et al. Mistral 7B (2023). 2023, arXiv:2310.06825. arXiv.org e-Print archive. http://arxiv.org/abs/2310.06825.
(16) Chiang, W.-L.; Zheng, L.; Sheng, Y.; Angelopoulos, A. N.; Li, T.; Li, D.; Zhang, H.; Zhu, B.; Jordan, M.; Gonzalez, J. E. et al. Chatbot Arena: An Open Platform for Evaluating llms by Human Preference. 2024, arXiv:2403.04132. arXiv.org e-Print archive. http://arxiv.org/abs/2403.04132.
(17) Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Adv. Neural Inf. Process Syst.* **2024**, *36*, 46595−46623.
(18) Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring Massive Multitask Language Understanding. 2020, arXiv:2009.03300. arXiv.org e-Print archive. http://arxiv.org/abs/2009.03300.
(19) Boiko, D. A.; MacKnight, R.; Gomes, G. Emergent Autonomous Scientific Research Capabilities of Large Language Models. 2023, arXiv:2304.05332. arXiv.org e-Print archive. http://arxiv.org/abs/2304.05332.
(20) Berthold, M. R.; Cebron, N.; Dill, F.; Gabriel, T. R.; Kötter, T.; Meinl, T.; Ohl, P.; Thiel, K.; Wiswedel, B. KNIME-the Konstanz information miner: version 2.0 and beyond. *AcM SIGKDD Explor. Newsl.* **2009**, *11*, 26−31.
(21) Goecks, J.; Nekrutenko, A.; Taylor, J. The Galaxy, T. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **2010**, *11*, R86.
(22) M Bran, A.; Cox, S.; Schilter, O.; Baldassari, C.; White, A. D.; Schwaller, P. Augmenting large language models with chemistry tools. *Nat. Mach. Intell.* **2024**, *6*, 1−11.
(23) Chase, H. LangChain, 2022. https://github.com/langchain-ai/langchain.
(24) Wolf, T.; Debut, L.; Sanh, V. et al. In *Transformers: State-of-the-Art Natural Language Processing*, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Stroudsburg, PA, USA, 2020; pp 38−45.
(25) Karpas, E.; Abend, O.; Belinkov, Y.; Lenz, B.; Lieber, O.; Ratner, N.; Shoham, Y.; Bata, H.; Levine, Y.; Leyton-Brown, K. et al. MRKL Systems: A Modular, Neuro-Symbolic Architecture That Combines Large Language Models, External Knowledge Sources and Discrete Reasoning. 2022, arXiv:2205.00445. arXiv.org e-Print archive. http://arxiv.org/abs/2205.00445.
(26) Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; Cao, Y. In *ReAct: Synergizing Reasoning and Acting in Language Models*, The Eleventh International Conference on Learning Representations, 2023.
(27) Landrum, G. et al. *RDKit: A Software Suite for Cheminformatics, Computational Chemistry, and Predictive Modeling*; Greg Landrum, 2013; Vol. *8*, p 5281.
(28) Daina, A.; Zoete, V. A boiled-egg to predict gastrointestinal absorption and brain penetration of small molecules. *ChemMedChem* **2016**, *11*, 1117−1121.
(29) Wildman, S. A.; Crippen, G. M. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 868−873.
(30) Ertl, P.; Rohde, B.; Selzer, P. Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *J. Med. Chem.* **2000**, *43*, 3714−3717.
(31) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **2012**, *4*, 90−98.
(32) Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminf.* **2009**, *1*, 1−11.
(33) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3−25.
(34) Brenk, R.; Schipani, A.; James, D.; Krasowski, A.; Gilbert, I. H.; Frearson, J.; Wyatt, P. G. Lessons learnt from assembling screening libraries for drug discovery for neglected diseases. *ChemMedChem* **2008**, *3*, 435−444.
(35) Baell, J. B.; Holloway, G. A. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.* **2010**, *53*, 2719−2740.
(36) Li, M.; Zhao, Y.; Yu, B.; Song, F.; Li, H.; Yu, H.; Li, Z.; Huang, F.; Li, Y. In *API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs*, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore, 2023; pp 3102−3116.
(37) Farn, N.; Shin, R. Tooltalk: Evaluating Tool-Usage in a Conversational Setting. 2023, arXiv:2311.10775. arXiv.org e-Print archive. http://arxiv.org/abs/2311.10775.
(38) Gentopia, 2023. https://github.com/Gentopia-AI/Gentopia.
(39) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; et al. PubChem 2023 update. *Nucleic Acids Res.* **2023**, *51*, D1373−D1380.
(40) Bento, A. P.; Hersey, A.; Félix, E.; Landrum, G.; Gaulton, A.; Atkinson, F.; Bellis, L. J.; De Veij, M.; Leach, A. R. An open source chemical structure curation pipeline using RDKit. *J. Cheminf.* **2020**, *12*, 1−16.
(41) Swanson, K.; Walther, P.; Leitz, J.; Mukherjee, S.; Wu, J. C.; Shivnaraine, R. V.; Zou, J. ADMET-AI: a machine learning ADMET platform for evaluation of large-scale chemical libraries. *Bioinformatics* **2024**, *40*, No. btae416.
(42) Fu, L.; Shi, S.; Yi, J.; Wang, N.; He, Y.; Wu, Z.; Peng, J.; Deng, Y.; Wang, W.; Wu, C.; et al. ADMETlab 3.0: an updated comprehensive online ADMET prediction platform enhanced with broader coverage, improved performance, API functionality and decision support. *Nucleic Acids Res.* **2024**, *52*, No. gkae236.
(43) Zhao, Y.; Mulder, R. J.; Houshyar, S.; Le, T. C. A review on the application of molecular descriptors and machine learning in polymer design. *Polym. Chem.* **2023**, *14*, 3325−3346.

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573

(44) Kunnakkattu, I. R.; Choudhary, P.; Pravda, L.; Nadzirin, N.; Smart, O. S.; Yuan, Q.; Anyango, S.; Nair, S.; Varadi, M.; Velankar, S. PDBe CCDUtils: an RDKit-based toolkit for handling and analysing small molecules in the Protein Data Bank. *J. Cheminf.* **2023**, *15*, 117.
(45) Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT understands, too. *AI Open* **2023**, No. 12.
(46) Joshi, R. P.; Gebauer, N. W.; Bontha, M.; Khazaieli, M.; James, R. M.; Brown, J. B.; Kumar, N. 3D-Scaffold: A deep learning framework to generate 3d coordinates of drug-like molecules with desired scaffolds. *J. Phys. Chem. B* **2021**, *125*, 12166−12176.
(47) McNaughton, A. D.; Bontha, M. S.; Knutson, C. R.; Pope, J. A.; Kumar, N. De novo Design of Protein Target Specific Scaffold-Based Inhibitors via Reinforcement Learning. 2022, arXiv:2205.10473. arXiv.org e-Print archive. http://arxiv.org/abs/2205.10473.
(48) Knutson, C.; Bontha, M.; Bilbrey, J. A.; Kumar, N. Decoding the protein−ligand interactions using parallel graph neural networks. *Sci. Rep.* **2022**, *12*, No. 7624.
(49) Joshi, R. P.; McNaughton, A.; Thomas, D. G.; Henry, C. S.; Canon, S. R.; McCue, L. A.; Kumar, N. Quantum mechanical methods predict accurate thermodynamics of biochemical reactions. *ACS Omega* **2021**, *6*, 9948−9959.
(50) Joshi, R. P.; Schultz, K. J.; Wilson, J. W.; Kruel, A.; Varikoti, R. A.; Kombala, C. J.; Kneller, D. W.; Galanie, S.; Phillips, G.; Zhang, Q.; et al. Ai-accelerated design of targeted covalent inhibitors for SARS-CoV-2. *J. Chem. Inf. Model.* **2023**, *63*, 1438−1453.
(51) Varikoti, R. A.; Schultz, K. J.; Kombala, C. J.; Kruel, A.; Brandvold, K. R.; Zhou, M.; Kumar, N. Integrated data-driven and experimental approaches to accelerate lead optimization targeting SARS-CoV-2 main protease. *J. Comput.-Aided Mol. Des.* **2023**, *37*, 339−355.
(52) Joshi, R. P.; Kumar, N. Artificial intelligence for autonomous molecular design: A perspective. *Molecules* **2021**, *26*, 6761.
(53) Jiang, D.; Wu, Z.; Hsieh, C.-Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J. Cheminf.* **2021**, *13*, 1−23.
(54) Goldsmith, B. R.; Esterhuizen, J.; Liu, J.; Bartel, C. J.; Sutton, C. Machine learning for heterogeneous catalyst design and discovery. *AIChE J.* **2018**, *64*, 2311−2323.
(55) Agrawal, A.; Choudhary, A. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Mater.* **2016**, *4*, No. 053208.

https://doi.org/10.1021/acsomega.4c08408
ACS Omega 2024, 9, 46563−46573