

# ChatMolData: A Multimodal Agent for Automatic Molecular Data Processing

Yi Yu, Huien Wang, Libin Zong, Bo Chen, Yaqin Li,\* and Xiaohui Yu\*

In recent years, the development of large language models (LLMs) has revolutionized various fields of natural science. However, their application in dealing with various molecular data remains constrained due to the reliance on single-modality inputs and outputs. ChatMolData, a novel LLM-based multimodal agent designed to handle diverse molecular data forms, including molecular databases, images, structure-specific files, and unstructured and structured documents, is introduced. ChatMolData integrates the capabilities of LLMs (e.g., GPT-4 and GPT-3.5) with the robust toolset that supports data retrieval, structuring, prediction, visualization, and search tasks. The agent employs a systematic cycle of reasoning and action to efficiently process complex tasks in molecular science. The evaluation demonstrates that ChatMolData achieves over 90% accuracy for 128 diverse tasks, effectively bridging the gap between experimenters and computational tools. Moreover, it is anticipated that the multimodal-agent strategy provides a pathway to expand data size and improve data accessibility, ultimately promoting molecular research and innovation.

The construction of LLM-powered agents presents a promising solution to these challenges by integrating LLMs with specialized tools and well-designed prompts that enhance their performance and reliability in specific tasks.<sup>[4–6]</sup>

In recent years, there has been considerable progress in applying LLM-based agents to the natural sciences, particularly in fields like chemistry, biology, and materials science.<sup>[7–10]</sup> These agents have shown potential in autonomous design of chemical experiments, automated multiomics analysis, high-fidelity materials knowledge retrieval, and so on.<sup>[11–15]</sup> However, most of these agents encompass a single modality, based solely on text-based inputs and outputs. This limitation creates a barrier for experimenters who need to process and analyze various types of data that go beyond simple textual information. In the field of

molecular science, these kinds of data are stored in various formats, including databases, PDF or comma-separated values (CSVs) files, molecular structure files (CIF, MOL, etc.), and images. Although a variety of application programming interfaces (APIs) and software programs are available to process different types of molecular data,<sup>[16–19]</sup> the steep learning curve and the need for programming expertise often hinder their effective use.<sup>[20]</sup> Hence, it gives rise to a disconnection between experimenters and algorithm developers.

Concurrently, there is an emerging research trend focused on extending these LLM-powered AI agents into the multimodal domain, which is referred to as large multimodal agents (LMAs) as well.<sup>[21–23]</sup> For instance, GPT-driver makes use of the OpenAI GPT-3.5 model to tackle the heterogeneous planner inputs, finally generating safe and comfortable driving trajectories.<sup>[24]</sup> Additionally, in order to improve comprehension of the video content, several video understanding agents are proposed to recognize objects, actions, and scenes from video data.<sup>[25–27]</sup> Last but not the least, some agents are designed to create and modify images, under the user's flexible requirements.<sup>[28–30]</sup> Accordingly, these LMA-related studies have introduced the concept of subtask tools designed to handle sophisticated data types, thereby enhancing the agents' capabilities to process diverse forms of information or data, particular visual or auditory data.<sup>[31]</sup>

Inspired by these advancements in multimodal LLM agents, we have developed a new LMA, named ChatMolData, designed to bridge the gap between experimenters and algorithm developers. With queries described in natural language and various types of molecular data as inputs, ChatMolData is shown to address a


## 1. Introduction

Large language models (LLMs) have achieved significant success in general artificial intelligence, demonstrating remarkable capabilities in tasks ranging from natural language processing to complex problem-solving across various domains.<sup>[1]</sup> However, in specialized fields such as medical and natural science, LLMs face certain limitations, including occasional inaccuracies or “hallucinations” when dealing with domain-specific knowledge.<sup>[2,3]</sup>

Y. Yu, H. Wang, L. Zong, X. Yu  
China Resources Pharmaceutical Research Institute (Shenzhen) Co., Ltd.  
Shenzhen 518100, China  
E-mail: yuxiaohui23@crpharm.com

B. Chen  
China Resources Pharmaceutical Group Ltd.  
Beijing 100029, China

Y. Li  
Faculty of Health and Social Sciences  
The Hong Kong Polytechnic University  
Hongkong, Hung Hom, China  
E-mail: yaqin0809.li@connect.polyu.hk

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202401089>.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202401089

variety of tasks in molecular data processing. These tasks include but are not limited to extracting and loading molecular data from natural language, structuring molecular data from literatures, visualizing molecular data mining results, and conducting sub-structure searches within molecular datasets. The performance of the agent is also evaluated and discussed under different variable conditions, which potentially provide inspiration for the LMAs design for data processing in various fields.

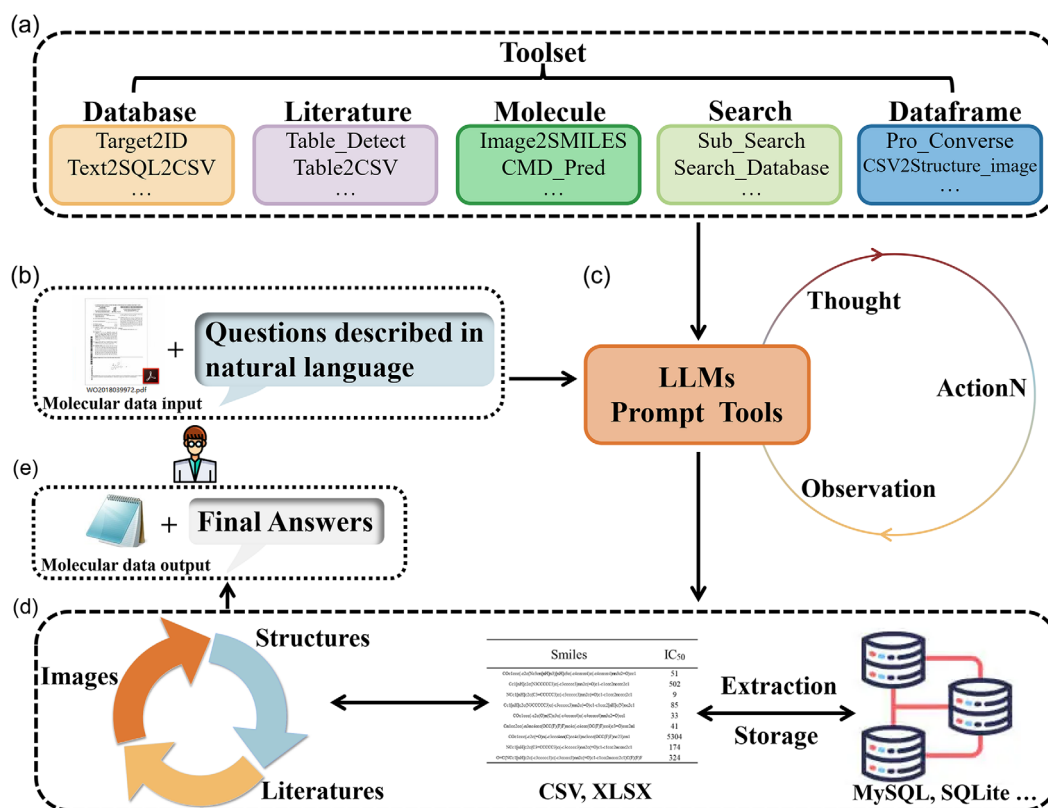
## 2. Results and Discussion

### 2.1. Design of ChatMolData

The ChatMolData agent is composed of three critical components: the LLM, a toolset, and a set of prompts. The LLM functions as the planner and the brain of the agent, similar to how a central processing unit operates in a computer system.<sup>[32–34]</sup> It is responsible for managing and evaluating processes, determining the appropriate tools to use, and making decisions based on the provided prompts and inputs. In this setup, we have utilized OpenAI's GPT-4 and GPT-3.5 as the LLM,<sup>[35,36]</sup> leveraging its advanced capabilities in reasoning, planning, and interacting with the environment to perform complex tasks beyond mere text

generation. As for the toolset, it is designed to handle a wide array of molecular data processing tasks, which consists of five modules: literature, molecule, database, search, and dataframe (depicted in Figure 1a). Each of these modules includes various tools specifically tailored to process different forms of molecular data, ensuring that the agent can proficiently extract, analyze, and manipulate data from diverse sources. As far as we know, one of the major advantages of agents is their ability to continuously iterate. The toolset of the agent can be modified or updated at any time based on user needs and technological advancements.<sup>[37]</sup> Here, we showcase the representative tools of the ChatMolData in Supporting Information, Note 1. Regarding prompts, it plays a crucial role in guiding the LLM's actions.<sup>[38]</sup> They serve as references, helping the LLM to think, reason, and decide on the appropriate course of action.<sup>[39,40]</sup> By structuring the LLM's interaction with the toolset and the data, prompts ensure that the agent can systematically approach and solve complex tasks. The details of the prompt are shown in Supporting Information, Note 2.

The workflow of ChatMolData draws inspiration from the ReAct<sup>[41]</sup> and MRKL<sup>[42]</sup> methodologies, integrating the Chain of Thought process with tool usage to address intricate problems. The agent processes tasks using a structured sequence of Thought, ActionN (N can be an integer starting from one,



**Figure 1.** Conceptual and schematic representation of ChatMolData. a) The toolset of ChatMolData includes database, literature, molecule, and search modules. Each module has specific tools to process different molecular data; b) Inputs from users are questions in natural language describing tasks and molecular files containing molecular structures or properties. c) In collaboration with LLMs, prompts and tools, ChatMolData promote the Reason + Act loop for dealing with complex tasks that require multistep processing, the loop includes Thought, ActionN (N stands for the order of actions) and Observation procedures. d) The functions of ChatMolData include conversion between different molecular data. e) Outputs of ChatMolData are final answers and molecular files from user's requests.

representing the sequence of actions), Action Input, and Observation. The architecture and operation mechanism are depicted in Figure 1. The input to the agent can be multiple types of molecular files coupled with a natural language task description provided by the user as illustrated in Figure 1b. The agent begins by engaging in the Thought step, where it contemplates the task at hand. It then proposes the first tool from the toolset to deal with the problem in the Action1 step, accompanied by Action Input describing the input of this action. Once the tool is executed, the result is returned as an Observation. After completing each loop of Thought, ActionN (N stands for the order of actions) and Observation, the agent reassesses the situation and determines the next loop as shown in Figure 1c. This could involve continuing the loop by invoking other tools or concluding the process by delivering the final answer along with the processed molecular data file. This iterative approach allows the ChatMolData agent to systematically break down complex tasks into manageable steps, ensuring accuracy and efficiency in its operations. Figure 1d demonstrates that ChatMolData facilitates the conversion between structured and unstructured data in molecular science, as well as molecular structures and properties. Finally, depicted in Figure 1e, the outputs of ChatMolData contain answers as well as files after molecular data processing. In the following sections, we will demonstrate that ChatMolData can execute four types of tasks for molecular data processing, including database-related, molecular data structuring, prediction and visualization, and structure searching.

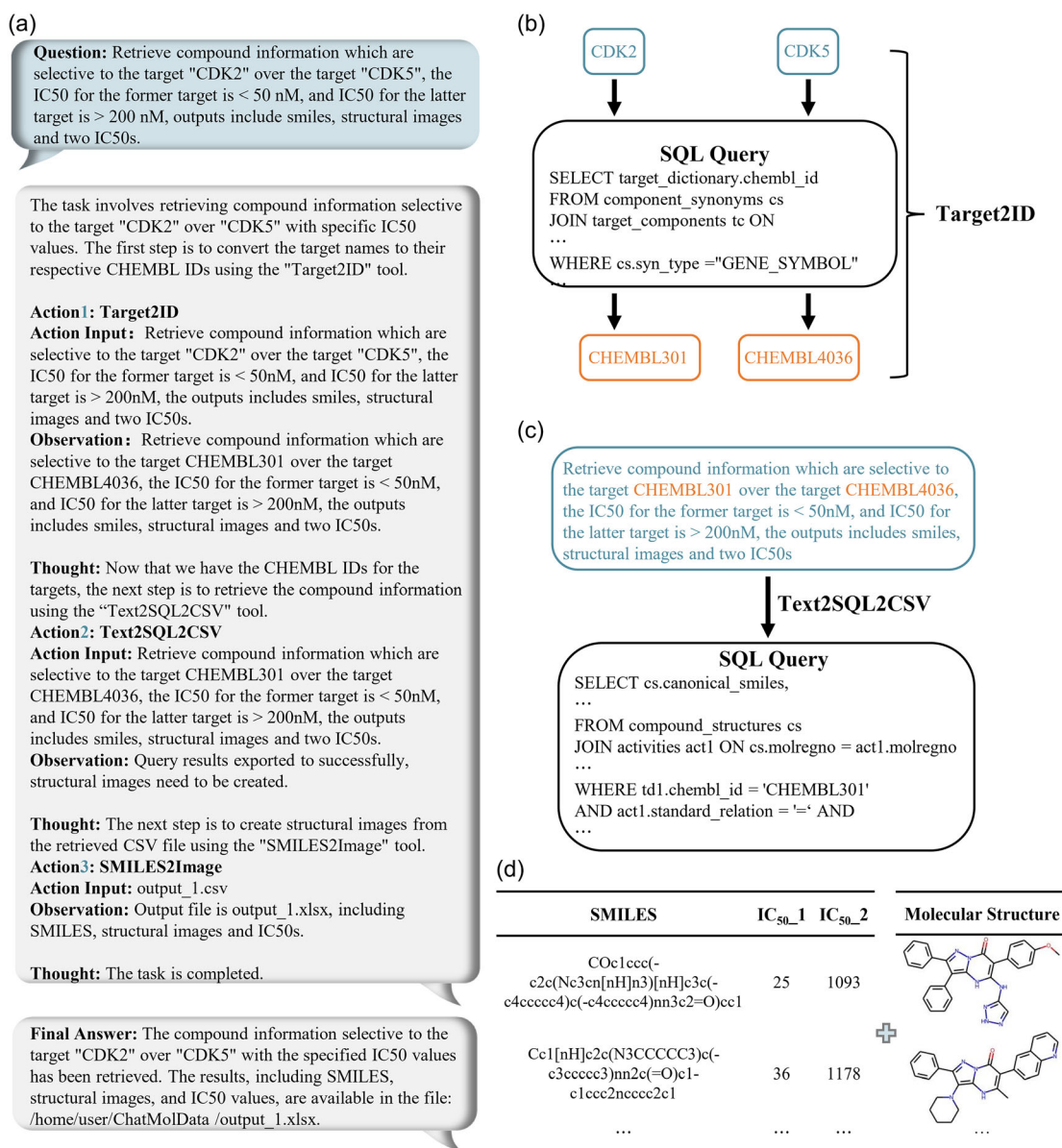
## 2.2. Database-Related Tasks

When a user, regardless of their programming or database experience, seeks to process molecular data from a database based on complex relationships, ChatMolData utilizes multiple tools within its Database module to achieve multistep data extraction or storage. **Figure 2** illustrates an example of retrieving molecular data from the ChEMBL database.<sup>[43,44]</sup> As shown in Figure 2a, upon receiving the question, ChatMolData begins its thought process. The agent first identifies that the retrieving task does not include a ChEMBL ID (It is a unique ID that has been assigned to compounds, targets, and assays in ChEMBL.) in the query. Consequently, the agent selects the Target2ID tool as Action1, attempting to replace the target abbreviation in the query with the corresponding ChEMBL ID. In Action1, the Action Input is the LLM-processed textual input of the retrieval request. Subsequently, as shown in Figure 2b, the Target2ID tool extracts the target abbreviation (CDK2 and CDK5 in this case) from the input above and then generates a natural language instruction to retrieve the ChEMBL IDs. This natural language command is converted into SQL language (create\_sql\_query\_chain function in LangChain is used here<sup>[45]</sup>) and then executed to obtain the ChEMBL IDs (CHEMBL301 and CHEMBL4036 here) corresponding to the target abbreviations. As for the Observation in the first cycle, the retrieval text with the target abbreviations is replaced by the ChEMBL IDs in the original query text, as illustrated in Figure 2c. In Action2, the agent proceeds to the second cycle, where, after the second thought process, it selects the Text2SQL2CSV tool. As shown in Figure 2c, the output of the Observation from the previous step is utilized as

the Action Input. The Text2SQL2CSV tool first converts the natural language into SQL language. (The detailed results of the above two steps of natural language to SQL language conversion are in Supporting Information, Note 3). After executing the SQL command, the Text2SQL2CSV tool generates the CSV file shown on the left side of Figure 2d, containing the simplified molecular input line entry system<sup>[46]</sup> (SMILES) and IC<sub>50</sub> information for two targets. In the Action3, during the Thought process, the agent notices that the generated CSV only includes the molecular SMILES representation, other than directly molecular structure. Therefore, the agent employs the CSV2Structure\_image tool to convert the SMILES into images representing molecular structures.<sup>[17]</sup> As depicted in Figure 2d, the generated images are saved in a new XLSX file, as well as the previous information in the last step. After executing these three Actions, the agent determines in the Thought process that the required molecular data has been generated, and thus it ends the cycle. At last, the agent provides the location of the molecular data file in the final answer, completing the entire automated reasoning process.

For experimenters, handling such molecular data extraction poses three main challenges: first, experimenters are accustomed to using target abbreviations as keywords for querying molecular structure and property data. However, in an SQL database, target abbreviations may lead to incorrect data retrieval; second, compared to single-target molecular activity data extraction, extracting multitarget selectivity data is more complicated. SQL language is the fastest and most convenient method for the retrieval of selectivity information,<sup>[47]</sup> yet most experimenters lack the foundational knowledge of SQL scripting; third, the molecular structures extracted from databases traditionally are represented as SMILES, a format that is not intuitive for visualizing structural differences, increasing the manual conversion workload for users. After the user inputs the natural language description of the molecular data extraction task, ChatMolData automatically overcomes these three challenges using the agent's built-in Target2ID, Text2SQL2CSV, and CSV2Structure\_image tools. As a result, users receive accurate and visually multimodal molecular data, significantly lowering the barriers to completing complex molecular data extraction tasks.

Besides operations in SQL databases, we also conducted data processing operations on a QM9,<sup>[48]</sup> a non-SQL database. As shown in Supporting Information, Note 4, structures and electronic properties are retrieved automatically. Regarding the diverse functions of database-related operations, ChatMolData receives natural language command and loads molecular data into the database. Specifically, as illustrated in Supporting Information, Note 5, the molecular information originally in a CSV file is inserted into the private database. In order to show the advantage of ChatMolData, the naïve GPT-4o model with the same structured prompts was evaluated for the database-related tasks. Consequently, GPT-4o cannot complete tasks, whether it is about SQL or non-SQL databases. The reason for the above results is that the GPT-4o is not suitable for processing extremely long texts, and it cannot directly connect to the SQL database to perform SQL operation. Accordingly, the above examples show that experimenters without programming or database background can use ChatMolData to perform a variety of database processing tasks using simple natural language commands, which facilitates the adaptation to big data for molecular science.



**Figure 2.** Example of a database-related task for retrieving molecular selectivity information between two targets. a) The question (blue dialog box) and output from ChatMolData, including 3 "reasoning + action" loops and a final answer (grey dialog boxes). b) Target2ID tool is used to convert target abbreviations (CDK2 and CDK5, in blue boxes) into ChEMBL IDs (CHEMBL301 and CHEMBL4036, in orange boxes), SQL queries (in black box) are generated and executed by Target2ID. c) Text2SQL2CSV tool converts the modified natural language query (in the orange box) into a SQL command (in black box), then the command is executed. d) The table on the left represents the generated CSV file including SMILES and IC<sub>50</sub> values by executing the command mentioned above, while CSV2Structure\_image tool converts SMILES into the molecular images on the right. Finally, the information in four columns is saved in an XLSX file (plus a sign in light blue representing the data merging process).

### 2.3. Molecular Data Structuring Tasks

In addition to performing database-related tasks, ChatMolData can utilize various tools within its Literature Module to carry out the collection of molecular information from unstructured data. As illustrated in Figure 3, the case demonstrates how ChatMolData extracts molecular structures and bioactivity data from a patent and ultimately stores them in a CSV file. In Figure 3a, the user's query is "Extract the molecular structures

and bioactivity information from 'WO2021252681.pdf and save them as a CSV file.'" ChatMolData initiates its reasoning process and identifies this task as a molecular data structuring task. The agent's Action1 involves using the Table\_Detection tool to retrieve all images containing table information from each page<sup>[49,50]</sup> (as shown in Figure 3a). Meanwhile, ChatMolData creates new folds to save the images where tables are on the consecutive pages, and the table column structures are same from each other. The output of the Observation includes the folder



(a)

**Question:** Extract the molecular structures and bioactivity information from WO2021252681.pdf and save them as a CSV file.

The task involves extracting need data and aggregating the information from a literature. The first step is to extract tables in the form of images.

#### Action1: Table\_Detect

**Action Input:** WO2021252681.pdf

**Observation:** Images are saved in /home/user/ChatMolData /fold\_1 and /home/user/ChatMolData/fold\_2.

**Thought:** The images regarding the tables has been extracted. Now, I will transfer the images into CSV files.

#### Action2: Table2CSV

**Action Input:** substructure.mol and CDK2\_mols\_pre.csv

**Observation:** CSVs are saved in /home/user/ChatMolData /fold\_1 and /home/user/ ChatMolData /fold\_2.

**Thought:** The CSVs regarding the tables has been extracted. Now, I will merge the tables that span multiple pages.

#### Action3: Y\_Concat

**Action Input:** /home/user/ChatMolData/fold\_1 and /home/user/ ChatMolData/fold\_2

#### Observation:

/home/user/ChatMolData/fold\_1/table\_xmerged.csv and /home/user/ ChatMolData/fold\_2/table\_xmerged.csv.

**Thought:** The tables spanning multiple pages have been merged. Now, I will merge the tables with the same columns.

#### Action4: X\_Concat

**Action Input:** /home/user/ChatMolData/table\_final.csv

**Observation:** The data has been extracted and aggregated.

**Final Answer:** The structures and bioactivities has been successfully executed. The data is saved in /home/user/ChatMol-Data/table\_final.csv.

(b)

Cpd. No.	Structure	<sup>1</sup> H NMR	Cpd. No.	IC <sub>50</sub>	Cpd. No.	IC <sub>50</sub>
1.001		<sup>1</sup> H NMR (400 MHz, DMSO-d <sub>6</sub> ) δ 7.73-7.65 (m, 2H), 7.59-7.55 (m, 3H), 7.47 (dd, J = 10.0, 3.0 Hz, 1H), 7.32 (d, J = 8.5 Hz, 1H), 6.58 (s, 1H), 3.67 (s, 3H).	1.001	>10 μM	1.014	+
1.002		<sup>1</sup> H NMR (500 MHz, DMSO-d <sub>6</sub> ) δ 12.28 (br s, 1H), 8.74 (s, 1H), 7.93 (d, J = 6.0 Hz, 1H), 7.63-7.59 (m, 1H), 7.49-7.41 (m, 5H), 7.30-7.28 (m, 2H), 7.17-7.14 (m, 1H), 7.01 (dd, J = 8.5, 1.0 Hz, 1H).	1.002	++	1.015	++++
1.003		<sup>1</sup> H NMR (500 MHz, DMSO-d <sub>6</sub> ) δ 12.44 (br s, 1H), 7.88-7.86 (m, 2H), 7.72-7.62 (m, 2H), 7.60-7.50 (m, 5H), 7.45-7.41 (m, 2H), 7.38 (d, J = 8.8 Hz, 1H), 6.74 (d, J = 2.4 Hz, 1H); LC-MS (ESI, m/z).	1.003	+	1.016	+
			1.004	+++	1.017	++++
			1.005	+	1.018	++++
			1.006	+	1.019	++++
			1.007	>10	1.020	++++

(c)

Cpd. No.	Structure	Cpd. No.	IC <sub>50</sub>
1.001	CN1C=CC=C(C=C2C(=CC1=O)C3=CC=CC=C3)C1	1.001	> 10 μM
1.002	C1=CC=C(C=C1)C2=C(C(=O)NC3=CC=CC=C3)SC4=NC=NC5=C4C=CS5	1.002	++
1.003	C1=CC=C(C=C1)C2=C(C(=O)NC3=CC=CC=C3)S(=O)(=O)C4=CC=CC=C4	1.003	+
...	...	...	...
+	+	+	+
Cpd. No.	Structure	Cpd. No.	IC <sub>50</sub>
...	...	1.008	+
1.008	CN(C)C1=C(C2=C(C(=CC(=C2)Cl)NC1=O)C3=CC=CC=C3	1.009	++++
1.009	COC1=C(C2=C(C(=CC(=C2)Br)NC1=O)C3=CC=CC=C3	...	...
...	...	...	...

(d)

Cpd. No.	Structure	Cpd. No.	IC <sub>50</sub>
1.001	COC1=C(C2=CC=CC=C2)C3=C(C(=CC(=C3)Cl)NC1=O	1.001	> 10 μM
1.002	C1=CC=C(C=C1)C2=C(C(=O)NC3=CC=CC=C3)SC4=C5C=CSC5=NC=N4	1.002	++
...	...	...	...
1.008	CN(C)C1=C(C2=C(C(=CC(=C2)Cl)NC1=O)C3=CC=CC=C3	1.008	+
1.009	COC1=C(C2=C(C(=CC(=C2)Br)NC1=O)C3=CC=CC=C3	1.009	++++
...	...	...	...

**Figure 3.** Example of a molecular data structuring task for exacting structures and bioactivity information, finally saving the structured data in a CSV file. a) The question (blue dialog box) and output from ChatMolData, including 4 “reasoning + action” loops and final answer (gray dialog boxes). b) Table\_Detect tool detects related tables including Cpd. No., Structure (SMILES) and IC<sub>50</sub> and then saves them in image format (The part in the dashed box is content to be extracted). c) Table2CSV tool interprets the table contents from images and next converts them into CSV files. Additionally, Y\_Concat tool is used to concatenate the extracted table information along the Y-axis (plus sign representing the processes), in order to merge crosspage table data. d) X\_Concat tool concatenates tables along the X-axis based on the shared information of molecular number between different tables.

names. In Action 2, the system reads the files from the saved folder, and the Table2CSV tool interprets the table contents in image format, ultimately converting them into CSV files stored in the same folder.<sup>[51,52]</sup> As shown in Figure 3b, before processing each table saved in an image file, the tables are categorized into two groups: one group without molecular structures and the other group containing molecular structures. For both groups, the process involves converting the images into CSV files. The key difference is that the former directly copies the information

from the image, while for the latter, the molecular structure images are processed using the DECIMER method, converting the molecular structure into SMILES format and saving it as a CSV file.<sup>[53]</sup> Finally, the observation output remains the folder name where the data is stored as well.

The above two operations realize the data extraction from the unstructured pdf file. However, data are scattered in different CSV files. In Action 3, the Y\_Concat tool is used to concatenate the extracted table information along the Y-axis, achieving the

structuring of cross-page table data. The observation confirms that tables across pages have been concatenated (as shown in Figure 3c). In the final step, the X\_Concat tool is applied to combine data from tables within different folders mentioned earlier, as shown in Figure 3d. The system identifies identical columns across different tables and concatenates them. Ultimately, the structured molecular data from the patent is stored in a single table as the output of the multimodal agent.

In the case earlier, the combined use of the Table\_Detect and Table2CSV tools allows the extraction of molecular structures and bioactivity data from each page of the patent document. Additionally, the sequential use of the Y\_Concat and X\_Concat tools ensures that the data extracted in earlier steps are consolidated across two dimensions, addressing challenges related to cross-page tables and dispersed data in different tables. Hence, using literatures as inputs, ChatMolData is capable of executing complex molecular data collection tasks and producing structured molecular data as output. To our knowledge, ChEMBL, the largest open-access database containing molecular structures and bioactivities, is manually curated and primarily sources its molecular data from publications (over 80 000 from papers and fewer than 3,000 from patents). In other words, the information in molecule-related patents has not been well collected. Therefore, ChatMolData has the potential to collect valuable structure–activity information from patents in bulk, complementing the ChEMBL database and scaling up the information for databases whenever we want.

As shown in Figure 3c,d, some of the extracted bioactivity data is represented using different numbers of “+” symbols to indicate levels of activity. ChatMolData can convert this single-column bioactivity description into a more detailed multicolumn format through natural language queries, using the Pro\_Converse tool (as shown in Supporting Information, Note 6). For instance, “++++” initially indicates an activity level of less than 10 nM. After receiving the query, the agent converts the data into three columns: one for the relation (“<”), one for the value (“10”), and one for the unit (“nM”). This task execution further enhances the compatibility of the structured data with the ChEMBL database, facilitating the subsequent data storage process. To our knowledge, the format of molecule-related patents is highly diverse, and sometimes the structure information is not shown within tables. Inspired by Morin’s work,<sup>[54]</sup> ChatMolData first uses the discriminative tool, Doc\_Seg, to analyze the contents of the unstructured file, extract the molecular structure, and save them in SMILES format, which distinguish between molecular structure, Markush structure, and background. Then, it uses the Image2SMILES tool to extract all the structures. The example mentioned above is demonstrated in Supporting Information, Note 7.

## 2.4. Prediction and Visualization Tasks

The two sections earlier, respectively, demonstrate cases of extracting molecular data from structured and unstructured data, and the final data is saved in CSV or XLSX files for researchers to use. In this section, we will demonstrate that ChatMolData is capable of analyzing molecular data in CSV or XLSX files by various tools within its DataFrame Module. As shown in Figure 4,

ChatMolData demonstrates how to predict molecular properties from a CSV file and ultimately visualize the property distribution. In Figure 4a, after the user submits the corresponding query, the ChatMolData selects the CMD\_preprocess tool as Action1, aiming to clean the input molecular data by removing invalid SMILES and converting molecular SMILES into canonical SMILES, as depicted in Figure 4b. In the next action, ChatMolData takes the cleaned molecular data as input and uses the CMD\_Pred tool to predict multiple properties for the molecules in the dataset. According to the query requirements, the predicted properties include molecular weight (MW), lipophilicity (AlogP), topological polar surface area (tPSA), and quantitative estimate of druglikeness (QED).<sup>[55]</sup> As shown in Figure 4c, the predicted properties expand the original table, and the new table is stored as a new file. At last, Action3 corresponds to the final query requirement, where the CMD\_plotting tool is used to display the distribution of multiple properties within the molecular set. As illustrated in Figure 4d, the LMA returns four histograms that depict MW, AlogP, tPSA, and QED, respectively.

Recently, LLMs were tested to directly predict molecular properties.<sup>[56]</sup> However, satisfactory prediction accuracy by LLMs requires a certain amount of molecular structure and real relevant property data to be provided in the prompt, which significantly compromises the convenience of property prediction workflows. In contrast, for ChatMolData, zero-shot prediction is realized due to the integration of external predicting tools. Moreover, ChatMolData offers three additional advantages in property prediction: first, multimodal Input Support: Unlike traditional methods limited to SMILES text input (e.g., CSV files), ChatMolData accepts diverse formats (MOL, SDF, PDB), eliminating tedious format conversions; second, adaptability to evolving needs: The framework allows seamless updates to its internal tools, enabling users to replace or integrate custom models or SOTA prediction algorithms as requirements or technologies advance; third, cross-module synergy: The prediction module operates synergistically with other modules (e.g., database storage, visualization) within unified workflows. In the following, work on extending the agent’s predictive capabilities to other types of molecular properties will be shown.

In recent years, various predictors based on deep neural networks have demonstrated outstanding performance in predicting bioactivity, toxicity, and material-related properties.<sup>[57,58]</sup> To expand the agent’s application in property prediction, several excellent models in property prediction have also been incorporated into ChatMolData as tools, such as continuous and data-driven descriptor<sup>[59]</sup> (CDDD) for prediction of biological activities, multi-task graph attention<sup>[60]</sup> for predicting toxicity properties, and Attentive FP<sup>[61]</sup> for predicting the electronic properties for molecular semiconductors. Examples of these property predictions are shown in the Supporting Information, Note 8–10. The results suggest that ChatMolData is capable of effectively utilizing deep learning models to accurately predict molecular properties as well as visualizing the property distributions.

## 2.5. Search Tasks

Besides molecular data extraction and prediction, structure search is substantially demanded by experimenters. For instance,

(a)

**Question:** For `mol_smiles.csv`, calculate the multiple properties of the molecules and the plot histogram showing distribution of different properties, properties include MW, ALogP, tPSA and QED.

The task involves calculating multiple properties of the molecules in the provided CSV file and then plotting histograms showing the distributions of these properties. I will start by preprocessing the CSV file using the 'CMD\_preprocess' tool.

**Action1: CMD\_preprocess**

**Action Input:** `mol_smiles.csv`

**Observation:** 1 of the SMILES is invalid. The updated file is `mol_smiles_cleaned.csv`

**Thought:** The CSV file has been preprocessed. Now, I will use the 'CMD\_Pred' tool to calculate multiple properties of the molecules and generate a new CSV file with these properties.

**Action2: CMD\_Pred**

**Action Input:** `mol_smiles_cleaned.csv`

**Observation:** The updated file is `mol_smiles_cleaned_pred.csv`.

**Thought:** The properties of the molecules have been calculated and saved in a new CSV file. Now, I will use the 'CMD\_plotting' tool to plot histograms showing the distributions of these properties.

**Action3: CMD\_plotting**

**Action Input:** `mol_smiles_cleaned_pred.csv`

**Observation:** Figures have been shown

**Thought:** The task is completed.

**Final Answer:** The multiple properties of the molecules in the provided CSV file have been calculated and histograms showing the distributions of these properties have been plotted. The final CSV file with the calculated properties is '`mol_smiles_cleaned_pred.csv`'.

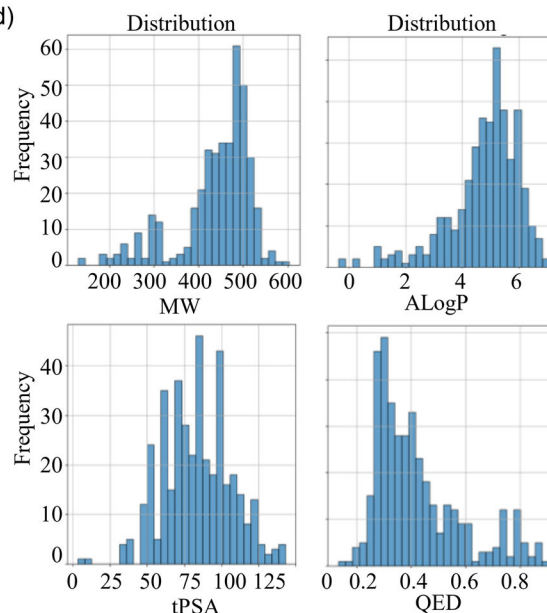
(b)

SMILES
<chem>COc1ccc(-c2c(Nc3cn[nH]n3)[nH]c3c(-c4ccccc4)c(-c4ccccc4)nn3c2=O)cc1</chem>
<chem>COQNCc1[nH]c2c(C3=CCCCC3)c(-c3ccccc3)nn2c(=O)</chem>
<chem>Cc1[nH]c2c(N3CCCCC3)c(-c3ccccc3)nn2c(=O)c1ccc2ncccc2c1</chem>
...

(c)

SMILES	MW	ALogP	tPSA	QED
<chem>COc1ccc(-c2c(Nc3cn[nH]n3)[nH]c3c(-c4ccccc4)c(-c4ccccc4)nn3c2=O)cc1</chem>	475	4.89	113.0	0.32
<chem>Cc1[nH]c2c(N3CCCCC3)c(-c3ccccc3)nn2c(=O)c1ccc2ncccc2c1</chem>	435	5.20	66.2	0.42
...	...	...	...	...

(d)



**Figure 4.** Example of a prediction and visualization task for calculating properties and showing the distribution of calculated properties. a) The question (blue dialog box) and output from ChatMolData, including three “reasoning + action” loops and final answer (gray dialog boxes). b) CMD\_preprocess tool is used to clean the molecular data by dropping the invalid SMILES and converting them into canonical SMILES. c) CMD\_Pred tool calculates the molecular weight (MW), lipophilicity (ALogP), topological polar surface area (tPSA), and Quantitative Estimate of Druglikeness (QED) and concatenates the properties and SMILES in a new CSV file. d) CMD\_plotting tool shows the property distributions by plotting the distributions of the properties mentioned above for the list of molecules.

molecular substructure search is beneficial for researchers in understanding the relationship between core structures and molecular properties, thereby facilitating molecular design. In this section, ChatMolData is utilized to perform this kind of task. In Figure 5a, the user provides a molecular substructure file (the structure has a position variation bond, representing multiple substructures<sup>[62]</sup>) and requests a substructure search within a CSV file (a molecular dataset containing SMILES and bioactivities). As shown in Figure 5b, upon receiving the task description and file information, the system first uses the Subsearch\_pre tool

to execute Action1 on the aforementioned substructure and molecular set file. This process enumerates all possible substructures (Figure 5c) while showing the bioactivity-related statistical information from the CSV file. In Action2, the Sub\_Search tool is employed to perform the substructure search task based on the two processed molecular files, ultimately returning the matching substructures (with matching parts highlighted) along with the corresponding activity data.<sup>[63]</sup> In this way, the user can search for multiple substructures simultaneously (as in this example, where substituent position variations are considered), improving



(a)

**Question:** For files `Substructure.mol` and `CDK2_mols.csv`, execute substructure search task.

The task requires performing the substructure search for the given file 'substructure.mol' and 'CDK2\_mols.csv'. The first step is to preprocess the csv file for the later search task.

**Action1:** `Subsearch_pre`

**Action Input:** "For `Substructure.mol` and `CDK2_mols.csv`, execute substructure search task."

**Observation:**

`substructure_pre.mol` and `CDK2_mols_pre.csv`

**Thought:** The preprocessing step has provided the necessary input files for the substructure search. Now, I will perform the substructure search using the provided files.

**Action2:** `Sub_search`

**Action Input:**

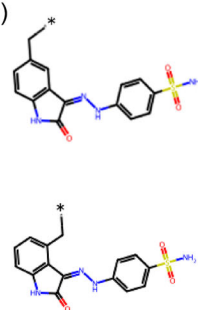
`Substructure_pre.mol` and `CDK2_mols_pre.csv`

**Observation:** Substructure search is completed

**Thought:** The task is completed.

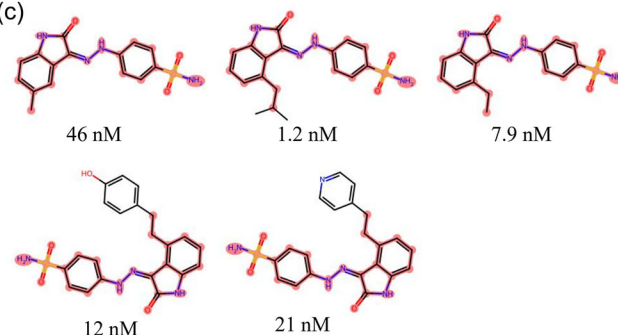
**Final Answer:** The substructure search for the file 'Substructure.mol' and 'CDK2\_mols.csv' has been successfully executed.

(b)



Statistics	Value
Mean	181
Std	238
Min	0.300
Max	992
25%	13.0
50%	60.0
75%	270

(c)



**Figure 5.** Example of a substructure search task for enumerating all the substructures and visualizing structure–property relationship after searching. a) The question (blue dialog box) and output from ChatMolData, including 2 “reasoning + action” loops and final answer (gray dialog boxes). b) `Subsearch_pre` tool enumerates all the substructures, table on the right side shows bioactivity-related statistical information in the molecular searching list. c) `Sub_Search` tool executes substructure search operation, showing matching structures (with matching parts highlighted) as well as the corresponding bioactivities.

search efficiency. Additionally, by referencing the activity statistical information from Action1, the user can gain insight into the structure–activity relationship of a particular substructure class based on the activity data of matching molecules, which contributes to the inspiration of molecular scaffold design.

Besides variation in molecular substituent positions, the diversity of substructure search tasks can be based on atom replacements and carbon chain length modifications as well by inputting different substructure variation files.<sup>[62]</sup> Beyond substructure searches, ChatMolData can also be used for full molecular structure searches. Supporting Information, Note 11, shows that the scope of structure searches is not confined to a molecular dataset in a CSV file but can also extend to direct searches within a SQL database by `Search_Database`. The output includes target information of the specific compound, as well as bioactivities, which can help researchers understand the interaction between specific molecular structures and multiple targets.

## 2.6. Evaluation

As described earlier, ChatMolData has demonstrated proficiency in handling four types of tasks: database-related, molecular data structuring, prediction and visualization, and search tasks. To

rigorously evaluate the performance of ChatMolData across these tasks, we designed 128 distinct tasks as benchmarks. These tasks varied across four dimensions, including the differences in the specified files, task subtypes, keywords, and task descriptions. The file differences refer to input file formats, such as molecular structures provided in either MOL or SDF formats. Task subtypes had been introduced previously; for example, in database-related tasks, some involved exporting molecular information from databases, while others required importing existing data files into a database. The keyword variations reflect subtle differences in task details; for instance, predicting molecular properties could target either bioactivity against a specific target or the physicochemical properties of the compounds. Based on these distinctions, each task category ultimately produced 8 subtasks, culminating in a total of 32 tasks. In order to evaluate the performance of the system under different expressions of the same task, for each task above, GPT-4 was further utilized to generate three additional queries to express tasks with different descriptions (examples are shown in Supporting Information, Note 12). This step was designed to examine whether ChatMolData could comprehend descriptions from users with different expression habits, which finally resulted in 128 tasks overall (as shown in Supporting Information, Note 12).



As mentioned in Section 2.2–2.4, current LLMs or agents in academia and industry lack robust multimodal processing capabilities for molecular data since a large part of molecular data, like structure image and databases, is unable to be read or connect directly. It indicated that existing LLMs and agents are not suitable for acting as the reference for comparison with ChatMolData. Inspired by the work on ChatMOF,<sup>[8]</sup> we evaluate the performance by varying the LLM within the agent framework. Three LLM configurations were compared: GPT-4, GPT-3.5-turbo, and GPT-3.5-turbo + prior knowledge (enhanced with detailed prompts describing complex tasks and tool usage). Given that existing automatic evaluation methods fail to capture the complexity of multimodal chemical data, all task outputs were evaluated by expert chemists.<sup>[9]</sup> The experts assigned each result a score of either “True” or “False.” A “True” score not only indicated that the output file and answer were satisfactory but also that the agent used the tools in an optimal and logical manner. Otherwise, the result is “False.” As shown in **Table 1**, ChatMolData, integrated with GPT-4, exhibits excellent performance. For the database-related, molecular data structuring, prediction and visualization, and search tasks, the accuracy was 90.6, 87.5, 96.9, and 93.8%, respectively, leading to an overall accuracy of 92.2%. These results indicate that in the vast majority of tasks, ChatMolData was able to effectively and logically utilize various tools, making it a superior multimodal agent for handling multiple types of molecular data and delivering satisfactory molecular files and answers.

When the LLM is GPT-3.5-turbo, the accuracy of the agent decreases, with 37.5, 78.1, 84.4, and 78.1% for database-related, molecular data structuring, prediction and visualization, and search tasks, respectively (the overall one is 69.5%). Regarding the significant difference in the performance for database-related tasks, it is found that in the text-to-SQL process, GPT-3.5-turbo failed to generate proper and complete SQL queries. The problem is in the process of executing the create\_sql\_query\_chain function because the performance of the function extremely depends on the LLM itself. In other words, GPT-3.5-turbo has intrinsic limitations to convert text to SQL language. Except the database-related tasks, the lower performance of GPT-3.5-turbo is attributed to two primary factors: query comprehension and reasoning ability. On the one hand, GPT-3.5-turbo’s inconsistency in handling varied user expressions across tasks hindered its performance. For example, when tasked with different phrasings of the same query, GPT-3.5-turbo failed to

consistently comprehend the meaning (i.e., it could understand Query\_2.2 and 2.4 describing a prediction and visualization task in Supporting Information, Note 12, but failed with Query\_2.1 and 2.3). The main reason for this result is that GPT-3.5-turbo has a weaker understanding of different expressions for a task than GPT-4. Therefore, it indicates that a more capable LLM enhances the agent’s ability to handle the same tasks expressed differently by various users, thereby improving the consistency and reproducibility of the agent. On the other hand, the capacity to handle complex tasks correlates strongly with the reasoning capabilities of the LLM, as reported in other studies.<sup>[8,15]</sup> To strengthen this hypothesis, we tested the performance of GPT-3.5-turbo with enhanced prior knowledge (GPT-3.5 + PK) by increasing the level of detail in the prompts (the accuracy was 84.4, 90.6, and 87.5% for molecular data structuring, prediction and visualization, and search tasks, respectively). For instance, when using GPT-4, the system will autonomously invoke the SubSearch tool to handle a substructure search task. In contrast, this only happens when the substructure search task is explicitly specified in the prompt, indicating the need to use the SubSearch tool when files include substructures and a list of molecules. Similar to the case of automated chemical experiments,<sup>[15]</sup> the reasoning ability of the agents benefits from the prior knowledge. However, one exception is the case of database-related tasks in which minimal improvement is shown with prior knowledge. It is noticed that the agent selects the correct tools and uses them in order. Nonetheless, GPT-3.5-turbo still struggles to accurately translate queries in natural language into SQL commands, which strengthens our hypothesis about LLMs’ intrinsic limitation for these kinds of tasks. Given the above analysis, the performance of ChatMolData heavily relies on the capabilities of the LLM.

### 3. Conclusion

This study presents the development and evaluation of ChatMolData, a multimodal AI agent designed to bridge the gap between experimenters and computational tools in molecular sciences. ChatMolData effectively processes various types of molecular data, including molecular images, structure-specific files, and unstructured and structured data format, thereby overcoming the limitations of single-modal LLM agents. Through natural language queries, users without programming skills can accomplish more than 100 tasks mentioned above with over 90% accuracy. Results of the evaluation by expert chemists indicated that the versatility and efficiency of ChatMolData are due to the integration of strong reasoning and perceptive LLM, elaborate prompt engineering, and various specialized molecular processing tools. Accordingly, our findings highlight that ChatMolData can be a transformative agent in molecular research, facilitating the expansion of molecular data size and convenience of molecular data utilization. As the field of multimodal AI continues to evolve, systems like ChatMolData offer a promising direction for the improvement of molecular data size and accessibility, ultimately promoting molecular research and innovation.

**Table 1.** Description of performance for four kinds of tasks using GPT-4, GPT-3.5-turbo, and GPT-3.5-turbo with prior knowledge (GPT-3.5 + PK), respectively.

LLM	Database <sup>a)</sup>	Structuring <sup>b)</sup>	Predication and <sup>c)</sup> visualization	Search <sup>d)</sup>	Overall
GPT-4	90.6%	87.5%	96.9%	93.8%	92.2%
GPT-3.5-turbo	37.5%	78.1%	84.4%	78.1%	69.5%
GPT-3.5 + PK	34.4%	84.4%	93.8%	87.5%	75.0%

<sup>a)</sup>Database-related tasks; <sup>b)</sup>Molecular data structuring tasks; <sup>c)</sup>Prediction and visualization tasks; <sup>d)</sup>Search tasks.

## 4. Experimental Section

Received: December 12, 2024

Revised: April 13, 2025

Published online:

**Agent Framework:** The framework of ChatMolData is based on LangChain,<sup>[61]</sup> which is a versatile framework designed to facilitate the development of applications powered by LLMs. By leveraging LangChain, developers can create dynamic, intelligent systems that go beyond simple text generation, incorporating logic, memory, and real-world interactions. LangChain's greatest strengths lies in its support for chain-of-thought (CoT) reasoning<sup>[64,65]</sup> and the seamless integration of external tools, such as APIs and databases. CoT reasoning allows models to perform step-by-step logical reasoning, improving accuracy and interpretability in tasks requiring complex decision-making. The ability to connect models to external tools further enhances their utility by allowing them to interact with real-time data and perform specialized tasks, such as querying molecular databases or running complex simulations. In ChatMolData, through LangChain, we integrated the three critical elements: LLMs, prompts, and tools for molecular data processing.

**Toolset:** In our implementation, the toolset can be divided into five modules: database, literature, molecule, search, and dataframe. Generally, the tools within the same module address similar types of tasks. For example, the tools in the database module are responsible for performing operations related to molecular database interactions. The tools of the ChatMolData in detail are described in Supporting Information Note 1.

## Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

## Acknowledgements

The authors gratefully acknowledge financial support from the Natural Science Foundation of Guangdong Province, China (grant no. 2024A1515011213).

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

**Yi Yu:** conceptualization (lead); formal analysis (lead); methodology (lead); conceptualization (lead); writing—original draft (lead); writing—review and editing (supporting). **Huien Wang:** data curation (supporting); formal analysis (equal); software (supporting). **Libin Zong:** formal analysis (supporting); writing—review and editing (equal). **Bo Chen:** writing—review and editing (supporting). **Yaqin Li:** supervision (lead); writing—review and editing (equal). **Xiaohui Yu:** conceptualization (equal); conceptualization (equal); writing—review and editing (equal).

## Data Availability Statement

The data that support the findings of this study are openly available in [ChatMolData] at [https://github.com/YiYuDL/ChatMolData], reference number [60].

## Keywords

cheminformatics, data mining, large language models, multimodal agents

- [1] S. Bubeck, V. Chadrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, *Sparks of Artificial General Intelligence: Early Experiments with GPT-4* **2023**.
- [2] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, *ACM Comput. Surv.* **2022**, 55, 1.
- [3] P. P. Ray, *Internet Things Cyber-Phys. Syst.* **2023**, 3, 121.
- [4] Z. Zhang, Y. Yao, A. Zhang, X. Tang, X. Ma, Z. He, Y. Wang, M. B. Gerstein, R. Wang, G. Liu, H. Zhao, *Igniting Language Intelligence: The Hitchhiker's Guide from Chain-of-Thought Reasoning to Language Agents* **2023**.
- [5] L. Wang, C. Ma, X. Feng, Z. Zhang, H.-r. Yang, J. Zhang, Z.-Y. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, J.-r. Wen, *Front. Comput. Sci.* **2024**, 18, 186345.
- [6] Q. Jin, Z. Wang, Y. Yang, Q. Zhu, D. Wright, T. Huang, W. J. Wilbur, Z. He, A. Taylor, Q. Chen, Z. Lu, *Agentmd: Empowering Language Agents for Risk Prediction with Large-Scale Clinical Tool Learning* **2024**.
- [7] G. P. Wellawatte, P. Schwaller, *Extracting Human Interpretable Structure-Property Relationships in Chemistry Using XAI and Large Language Models* **2023**.
- [8] Y. S. Kang, J. Kim, *Nat. Commun.* **2024**, 15, 4705.
- [9] A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, P. Schwaller, *Nat. Mach. Intell.* **2023**, 6, 525.
- [10] Z. Zheng, O. Zhang, H. L. Nguyen, N. Rampal, A. H. Alawadhi, Z. Rong, T. Head-Gordon, C. Borgs, J. T. Chayes, O. M. Yaghi, *ACS Cent. Sci.* **2023**, 9, 2161.
- [11] C. Yuan, E. Hsieh, C. H. Chou, J. Riebesell, *LLaMP: Large Language Model Made Powerful for High-Fidelity Materials Knowledge Retrieval and Distillation* **2024**.
- [12] Y. Qu, K. Huang, H. Cousins, W. A. Johnson, D. Yin, M. Shah, D. Zhou, R. Altman, M. Wang, L. Cong, *bioRxiv* **2024**, 2024.
- [13] H. Liu, Y. Li, J. Jian, Y. Cheng, J. Lu, S. Guo, J. Zhu, M. Zhang, M. Zhang, H. Wang, *Toward a Team Of AI-Made Scientists for Scientific Discovery from Gene Expression Data* **2024**.
- [14] Y. Ma, Z. Gou, J. Hao, R. Xu, S. Wang, L. Pan, Y. Yang, Y. Cao, A. Sun, H. Awadalla, W. Chen, in *ACL* **2024**, pp. 15701-15736.
- [15] D. A. Boiko, R. MacKnight, B. Kline, G. Gomes, *Nature* **2023**, 624, 570.
- [16] DeepChem, <https://github.com/deepchem/deepchem>, (accessed: February 2024).
- [17] RDKit, [www.rdkit.org](http://www.rdkit.org), (accessed: February 2024).
- [18] Open Babel, <https://github.com/openbabel/openbabel>, (accessed: February 2024).
- [19] rxn4Chemistry, <https://github.com/rxn4chemistry/rxn4chemistry>, (accessed: February 2024).
- [20] M. Miyagi, E. Kiesel, K. Neumbo, T. Nakazawa, *Anal. Chem.* **2024**, 96, 3077.
- [21] S. Asif Imran, M. N. H. Khan, S. Biswas, B. Islam, *LLaSA: Large Multimodal Agent for Human Activity Analysis Through Wearable Sensors* **2024**.
- [22] J. Xie, Z. Chen, R. Zhang, X. Wan, G. Li, *Large Multimodal Agents: A Survey* **2024**.
- [23] Z. Zheng, N. Rampal, T. J. Inizan, C. Borgs, J. T. Chayes, O. M. Yaghi, *Nat. Rev. Mater.* **2025**, 10, 369.
- [24] J. Mao, Y. Qian, J. Ye, H. Zhao, Y. Wang, *GPT-Driver: Learning to Drive with GPT* **2023**.
- [25] J. Wang, D. Chen, C. Luo, X. Dai, L. Yuan, Z. Wu, Y. G. Jiang, *ChatVideo: A Tracklet-Centric Multimodal and Versatile Video Understanding System* **2023**.

- [26] Z. Yang, G. Chen, X. Li, W. Wang, Y. Yang, *DoraemonGPT: Toward Understanding Dynamic Scenes with Large Language Models* **2024**.
- [27] D. Gao, L. Ji, L. Zhou, K. Q. Lin, J. Chen, Z. Fan, M. Z. Shou, *AssistGPT: A General Multi-Modal Assistant that can Plan, Execute, Inspect, and Learn* **2023**.
- [28] S. Li, R. Wang, C. J. Hsieh, M. Cheng, T. Zhou, *MuLan: Multimodal-LLM Agent for Progressive and Interactive Multi-Object Diffusion* **2024**.
- [29] Z. Yang, L. Li, J. Wang, K. Lin, E. Azarnasab, F. Ahmed, Z. Liu, C. Liu, M. Zeng, L. Wang, *MM-REACT: Prompting ChatGPT For Multimodal Reasoning and Action* **2023**.
- [30] W.-G. Chen, I. Spiridonova, J. Yang, J. Gao, C. Li, *LLaVA-Interactive: An All-In-One Demo for Image Chat, Segmentation, Generation and Editing* **2023**.
- [31] B. Jiang, Y. Xie, X. Wang, Y. Yuan, Z. Hao, X. Bai, W. J. Su, C. J. Taylor, T. Mallick, *Towards Rationality in Language and Multimodal Agents: A Survey* **2024**.
- [32] H. Zhang, W. Du, J. Shan, Q. Zhou, Y. Du, J. B. Tenenbaum, T. Shu, C. Gan, *Building Cooperative Embodied Agents Modularly with Large Language Models* **2023**.
- [33] A. Zhang, Y. Chen, H. Li, Y. Deng, X. Wang, T.-S. Chua, presented at Pro. of the 47th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Washington, DC, USA **2024**.
- [34] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, R. Zheng, X. Fan, X. Wang, L. Xiong, Q. Liu, Y. Zhou, W. Wang, C. Jiang, Y. Zou, X. Liu, Z. Yin, S. Dou, R. Weng, W. Cheng, Q. Zhang, W. Qin, Y. Zheng, X. Qiu, X. Huan, T. Gui, *The Rise and Potential of Large Language Model Based Agents: A Survey* **2023**.
- [35] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat, *GPT-4 Technical Report* **2023**.
- [36] ChatGPT, <https://openai.com/index/chatgpt/>, (accessed: April 2024).
- [37] S. Liu, Y. Lu, S. Chen, X. Hu, J. Zhao, Y. Lu, Y. Zhao, *DrugAgent: Automating AI-Aided Drug Discovery Programming Through LLM Multi-Agent Collaboration* **2024**.
- [38] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, *ACM Comput. Surv.* **2021**, 55, 1.
- [39] J. White, S. Hays, Q. Fu, J. Spencer-Smith, D. C. Schmidt, *ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design* **2023**.
- [40] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. C. Schmidt, presented at Proc. of the 30th Conf. on Pattern Languages of Programs, Monticello, IL, USA **2023**.
- [41] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, *ReAct: Synergizing Reasoning and Acting in Language Models* **2022**.
- [42] E. D. Karpas, O. Abend, Y. Belinkov, B. Lenz, O. Lieber, N. Ratner, Y. Shoham, H. Bata, Y. Levine, K. Leyton-Brown, D. Muhlgay, *MRKL Systems: A Modular, Neuro-Symbolic Architecture That Combines Large Language Models, External Knowledge Sources and Discrete Reasoning* **2022**.
- [43] B. Zdravil, E. Felix, F. Hunter, E. J. Mannes, J. Blackshaw, S. Corbett, M. de Veij, H. Ioannidis, D. M. Lopez, J. F. Mosquera, M. P. Magarinos, *Nucleic Acids Res.* **2023**, 52, D1180.
- [44] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, J. P. Overington, *Nucleic Acids Res.* **2011**, 40, D1100.
- [45] create\_sql\_query\_chain, [https://api.python.langchain.com/en/latest/chains/langchain.chains.sql\\_database.query.create\\_sql\\_query\\_chain.html](https://api.python.langchain.com/en/latest/chains/langchain.chains.sql_database.query.create_sql_query_chain.html), (accessed: February 2024).
- [46] D. Weininger, *J. Chem. Inf. Comput. Sci.* **1988**, 28, 31.
- [47] Schema Questions and SQL Examples, <https://chembl.gitbook.io/chembl-interface-documentation/frequently-asked-questions/schema-questions-and-sql-examples>, (accessed: February 2024).
- [48] N. J. Williams, L. Kabalan, L. Stojanović, V. Zolyomi, E. O. Pyzer-Knapp, *Hessian QM9: A Quantum Chemistry Database of Molecular Hessians In Implicit Solvents* **2024**.
- [49] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, *Deformable DETR: Deformable Transformers for End-To-End Object Detection* **2020**.
- [50] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, in *In European Conf. on Computer Vision*, Springer Inter. Publishing, Cham **2020**, pp. 213–229.
- [51] B. Smock, *PubTables-1M: Towards A Universal Dataset and Metrics for Training and Evaluating Table Extraction Models* **2021**.
- [52] B. Smock, R. Pesala, R. Abraham, *GriTS: Grid Table Similarity Metric for Table Structure Recognition* **2022**.
- [53] K. Rajan, A. Zielesny, C. Steinbeck, *J. Cheminform.* **2020**, 12, 65.
- [54] L. Morin, V. Weber, G. I. Meijer, F. Yu, P. W. Staar, *Nat. Commun.* **2024**, 15, 6532.
- [55] RDKit QED module, <https://www.rdkit.org/docs/source/rdkit.Chem.QED.html#rdkit.Chem>, (accessed: February 2024).
- [56] X. Cai, H. Lai, X. Wang, L. Wang, W. Liu, Y. Wang, Z. Wang, D. Cao, X. Zeng, *Methods* **2024**, 222, 133.
- [57] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoese, H. Schopmans, T. Sommer, P. Friederich, *Commun. Mater.* **2022**, 3, 93.
- [58] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, V. Pande, *Chem. Sci.* **2017**, 9, 513.
- [59] R. Winter, F. Montanari, F. Noé, D. A. Clevert, *Chem. Sci.* **2018**, 10, 1692.
- [60] Z. Wu, D. Jiang, J. Wang, C. Y. Hsieh, D. Cao, T. Hou, *J. Med. Chem.* **2021**, 64, 6924.
- [61] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, M. Zheng, *J. Med. Chem.* **2020**, 63, 8749.
- [62] Intro to the molecule enumerator, <https://greglandrum.github.io/rdkit-blog/posts/2021-05-13-intro-to-the-molecule-enumerator.html>, (accessed: February 2024).
- [63] Generalized substructure search, <https://greglandrum.github.io/rdkit-blog/posts/2021-08-03-generalized-substructure-search.html>, (accessed: April 2024).
- [64] LangChain, <https://github.com/langchain-ai/langchain>, (accessed: February 2024).
- [65] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *Adv. Neural Info. Process. Syst.* **2022**, 35, 24824.