Application of Self-Evolving AI Agents in Chemical Research: A Novel Intelligent Assistance System

Kangyong Ma[1*]

1. College of Physics and Electronic Information Engineering,Zhejiang Normal University, Jinhua City 321000, China.

E-mail address: kangyongma@outlook.com(K.y.Ma)； kangyongma@gmail.com

ORCID: 0000-0001-6948-2560

Abstract

This work utilizes collected and organized instructional data from the field of chemical science to fine-tune mainstream open-source large language models. To objectively evaluate the performance of the fine-tuned models, we have developed an automated scoring system specifically for the chemistry domain, ensuring the accuracy and reliability of the evaluation results. Building on this foundation, we have designed an innovative chemical intelligent assistant system. This system employs the fine-tuned Mistral Nemo model as one of its primary models and features a mechanism for flexibly invoking various advanced models. This design fully considers the rapid iteration characteristics of large language models, ensuring that the system can continuously leverage the latest and most powerful AI capabilities.A major highlight of this system is its deep integration of professional knowledge and requirements from the chemistry field. By incorporating specialized functions such as molecular visualization, SMILES string processing, and chemical literature retrieval, the system significantly enhances its practical value in chemical research and applications. More notably, the system possesses autonomous evolution capabilities. Through carefully designed mechanisms for knowledge accumulation, skill acquisition, performance evaluation, and group collaboration, the system can continuously optimize its professional abilities and interaction quality. This dynamic adaptive feature enables the system to evolve autonomously, breaking through the

inherent static limitations of traditional AI systems.

Key words : large language models; fine-tuning; chemistry; autonomous evolution

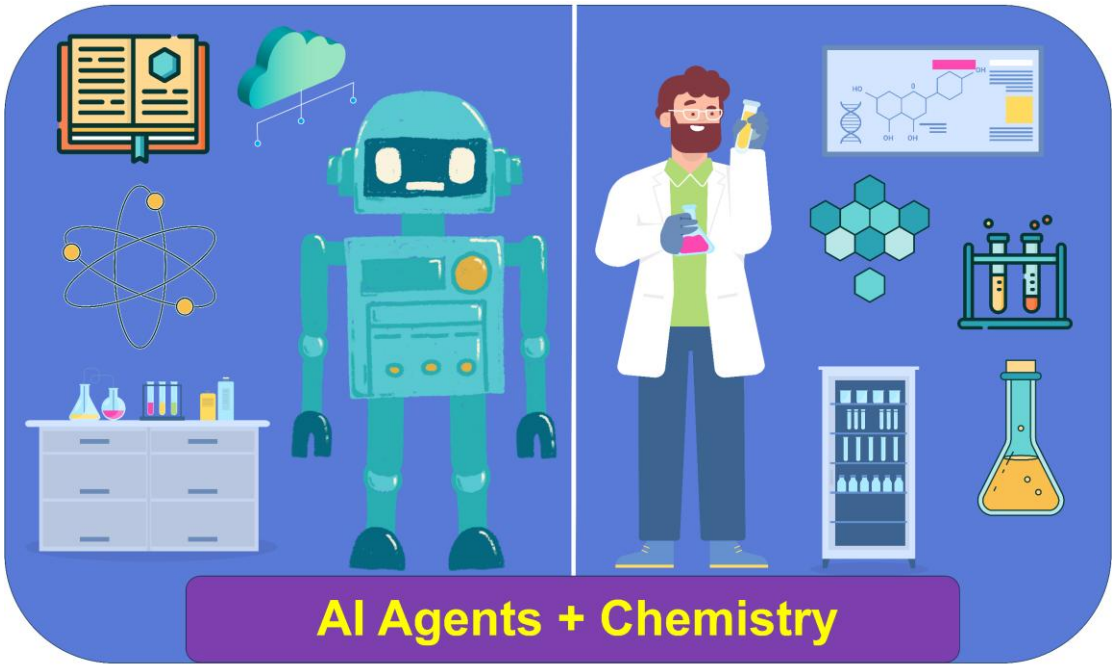

Fig TOC

1. *Introduction*

Large Language Models (LLMs) stand out as one of the most noteworthy achievements in the field of artificial intelligence in recent years and represents a crucial direction for the development of Artificial General Intelligence (AGI)[1,2]. Since the introduction of ChatGPT and GPT-4o, Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) have attracted significant interest due to their versatile abilities in understanding, reasoning, and generating content[3]. However, the current state of this technology still presents significant deficiencies and imbalances, including persistent illusions, misaligned values, weak specialization, and the black box effect[2]. In this scenario, how to apply Large Language Models (LLMs) to different professional fields has become a current research hotspot.

Fine-tuning has a significant effect on improving the performance of LLM in specific application scenarios, which lays the foundation for LLM to further promote scientific progress in various fields[4,5]. For example, research by Ouyang et al. (2022), Wei et al. (2021), and Sanh et al. (2021) demonstrates that fine-tuning language models on a specific set of tasks significantly enhances their ability to understand and execute instructions[6-8]. This method not only reduces the reliance on large datasets but also improves the generalization capabilities of the models. Given the scale of LLMs, a common fine-tuning strategy currently involves adjusting a limited number of parameters while keeping the rest fixed[9]. This technique, known as Parameter-Efficient Fine-Tuning (PEFT), selectively tunes a small subset of parameters. PEFT has also gained interest beyond NLP, particularly in the CV community, for fine-tuning large-parameter visual models like Vision Transformers (ViTs), diffusion models, and visual-language models[4].

However, fine-tuning large models still has some drawbacks. For example, this method requires substantial computational resources and data. Fine-tuning large models is also prone to overfitting on small-scale datasets and cannot accurately reflect potential risks (e.g., "hallucinations"), which may introduce latent hazards. Additionally, it cannot update its knowledge base in real time[10]. The primary reasons

for these drawbacks are that both pre-trained large models and fine-tuned large models use parameter memory to construct a parameterized implicit knowledge base[11]. Hybrid models that combine parametric memory and non-parametric (i.e., retrieval-based) memory can address some of these issues[12-14]. The Retrieval-Augmented Generation (RAG) technique improves the accuracy and reliability of hybrid model generation by integrating knowledge from external databases (non-parametric memory), especially for knowledge-intensive tasks. This approach also allows for continuous knowledge updates and the integration of domain-specific information. RAG synergizes the intrinsic knowledge of large language models with the extensive dynamic repositories of external databases[15].

Furthermore, with the continuous development of LLMs, they are seen as potential sparks for Artificial General Intelligence (AGI), providing hope for the construction of general AI agents[16]. Currently, AI agents are considered a crucial step towards achieving AGI, encompassing the potential for a wide range of intelligent activities[17-19].In many real-world tasks, the capabilities of agents can be enhanced by constructing multiple cooperative agents[20]. Studies have shown that multi-agent systems help encourage divergent thinking[21] (Liang et al., 2023), improve factuality and reasoning abilities[22] (Du et al., 2023), and provide verification[23] (Wu et al., 2023). These features have garnered widespread attention. Currently, the general frameworks for constructing LLM applications with multiple agents include AutoGen[37], crewAI[38],Langchain[39] and others. Intelligent agents based on large language models (LLMs) are increasingly permeating various aspects of human production and daily life. However, designing artificial intelligence agents with self-evolution capabilities has become a current research hotspot. For example, Li et al.[24] proposed an evolutionary framework for agent evolution and arrangement called EvoluaryAgent. Qian et al.[25] proposed a general strategy for inter-task agent self-evolution based on Investigation-Consolidation-Exploitation(ICE).

These artificial intelligence technologies will provide a new paradigm for scientific research and open new avenues for scientific innovation, thereby significantly accelerating the pace of scientific discoveries. The close collaboration

between artificial intelligence technologies and scientists heralds the advent of a new era of scientific exploration and technological breakthroughs[26,27].

In recent years, despite the rapid development of artificial intelligence technology, especially the emergence of large language models, its application in the field of chemistry has not yet been widely popularized. As an important productivity tool, artificial intelligence not only improves work efficiency but also provides a new paradigm for scientific research. For chemistry, a discipline with a long history, how to combine with this advanced productivity tool to breathe new life into the field has become an important topic facing the new generation of chemists. This research aims to address this challenge by developing a dedicated intelligent assistance system for the field of chemistry through the integration of cutting-edge AI technologies.Specifically, we first collected and organized a large amount of data from the field of chemical science to fine-tune mainstream open-source large language models. Secondly, we designed a set of evaluation systems specifically for the chemistry field to detect the performance of the fine-tuned models and select the best-performing model from them. On this basis, we developed an AI assistant for the chemistry field with autonomous evolution capabilities. This system integrates multi-agent architecture, retrieval-augmented generation (RAG) technology, online search functionality, dynamic learning and evolution mechanisms, and an interactive user interface. It not only provides an innovative platform for chemical research and education but also offers valuable research opportunities for exploring multi-agent collaboration and evolution mechanisms in complex systems.By fusing traditional chemical knowledge with cutting-edge AI technology, this system is expected to promote innovative development in the field of chemistry and provide new ideas and tools for solving current scientific and engineering challenges.

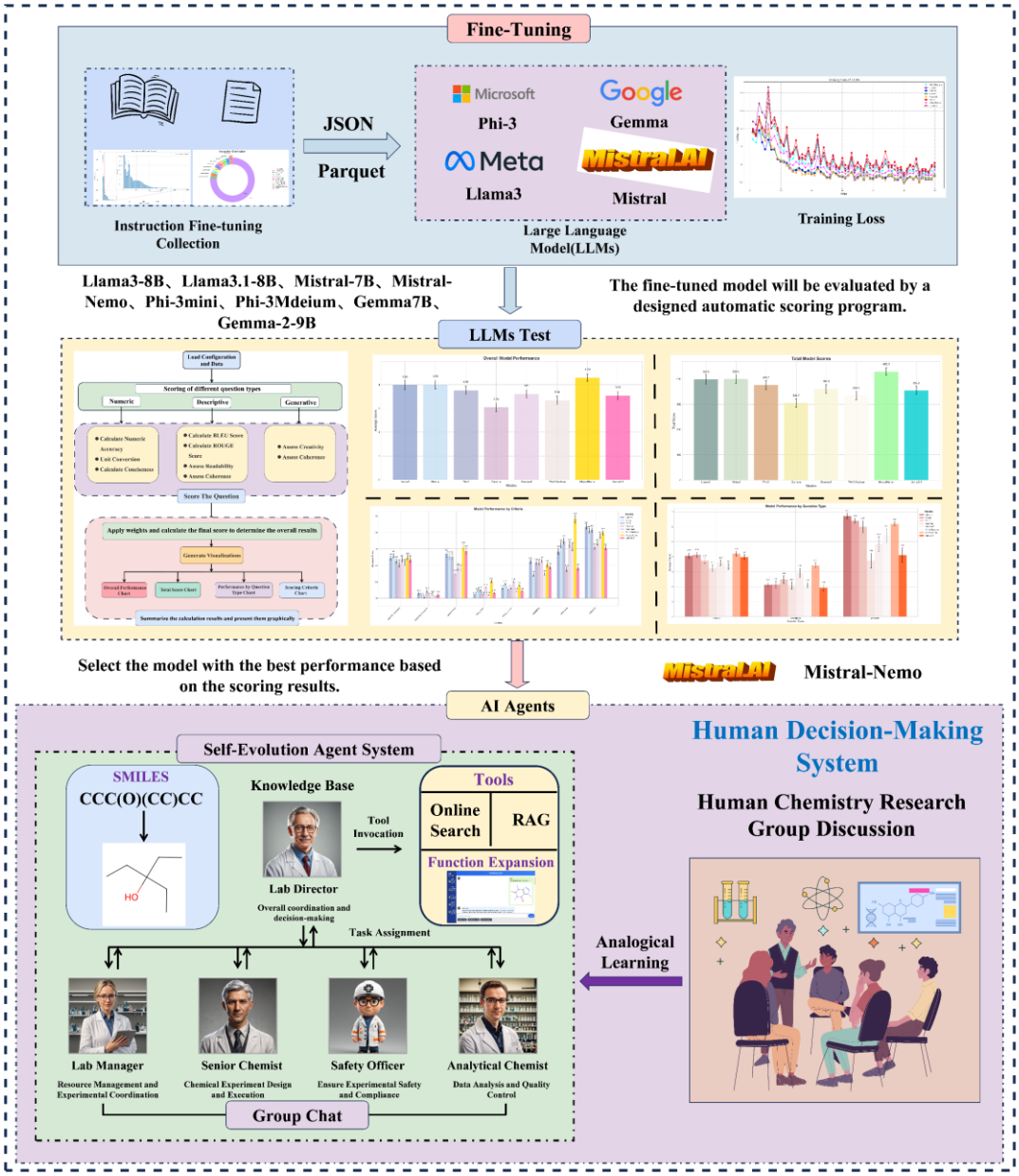

Fig1. Research Process

2. Related Work

2.1 Fine-tuning LLMs for Applications in the Field of Chemistry

In recent years, with the rapid development of artificial intelligence technology, Large Language Models (LLMs) have been increasingly applied in the field of chemical sciences. Through fine-tuning for specific chemical tasks, these models have demonstrated remarkable potential, bringing new perspectives and methods to chemical research. Currently, significant progress has been made in chemical science research using fine-tuned large language models, covering various aspects from material design to drug discovery. These studies not only showcase the exceptional ability of LLMs in handling complex chemical problems but also provide innovative approaches to addressing long-standing chemical challenges.

For example, Kevin Maik Jablonka et al. [45] fine-tuned the large language model GPT-3 to perform various tasks in chemistry and materials science, including properties of molecules and materials, as well as chemical reaction outcomes. Zikai Xie et al. [46] demonstrated the effectiveness of fine-tuned GPT-3 in predicting electronic and functional properties of organic molecules. Shifa Zhong et al. [47] developed quantitative structure-activity relationship (QSAR) models for water pollutant activity/properties by fine-tuning GPT-3 models. Seongmin Kim et al. [48] evaluated the effectiveness of pre-trained and fine-tuned large language models (LLMs) in predicting the synthesizability of inorganic compounds and selecting synthetic precursors. Results showed that fine-tuned LLMs performed comparably, and sometimes superiorly, to recent custom machine learning models in these tasks, while requiring less user expertise, cost, and time to develop.

These research findings conclusively demonstrate that fine-tuning LLMs can significantly enhance their application breadth and effectiveness in the field of chemical sciences. This approach not only provides powerful tools for chemical research but also promises to accelerate innovation in chemical sciences, offering new ideas and methods for solving complex chemical problems. As technology continues to advance, we can anticipate that fine-tuned LLMs will play an increasingly

important role in the field of chemical sciences, driving chemical research towards deeper and more precise directions.

2.2 AI agents in the field of Chemistry

Although large language models (LLMs) have demonstrated excellent performance in tasks across multiple domains, they face challenges in chemistry-related problems and lack the ability to access external knowledge sources, limiting their practicality in scientific applications. To address these deficiencies, researchers have conducted relevant explorations.

For example, Kevin Maik Jablonka et al. [49] developed ChemCrow, an LLM chemical agent designed to complete chemistry tasks such as organic synthesis, drug discovery, and materials design. By integrating multiple expert-designed chemical tools and using GPT-4 as the LLM, they enhanced the performance of LLMs in the field of chemistry and demonstrated new capabilities.Daniil A. Boiko et al. [50] reported on Coscientist, a GPT-4-powered artificial intelligence system capable of autonomously designing, planning, and executing complex scientific experiments. Coscientist leverages large language models combined with tools such as internet searches, document retrieval, code execution, and experimental automation.Andrew D. McNaughton et al. [51] introduced a system called CACTUS (Chemistry Agent Connecting Tool-Usage to Science), which is an intelligent agent based on large language models (LLMs) designed to enhance advanced reasoning and problem-solving capabilities in the fields of chemistry and molecular discovery by integrating cheminformatics tools.

These research findings demonstrate that AI Agents, by expanding the functionality of large language models, enable their more extensive application in the field of chemistry.

3. LLMS Fine-tuning Methods

3.1 Dataset

During the adjustment of the LLM, various datasets related to chemical sciences were utilized. The fine-tuning data for this work comes from the datasets listed in Table 1, with a total of 1.72 million fine-tuning instructions collected and organized. Based on this, two different types of instructions were divided for the fine-tuning training of different large models.

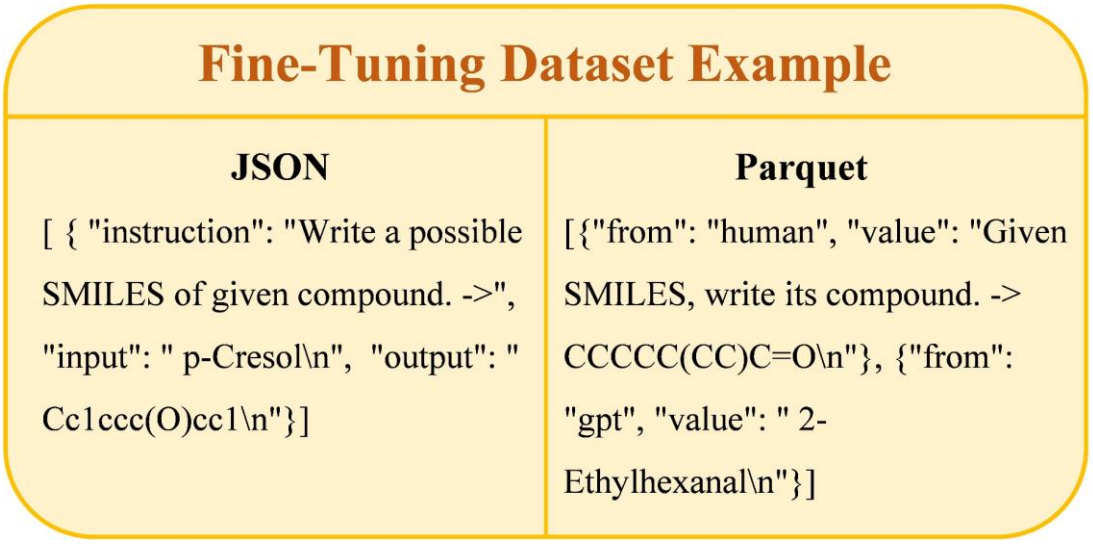

Figure 2. Example of fine-tuned data

Figures.3 and Figures.4 show the distribution of output character lengths for the instruction dataset and the usage frequency and types of the 20 most commonly used instructions in this work.

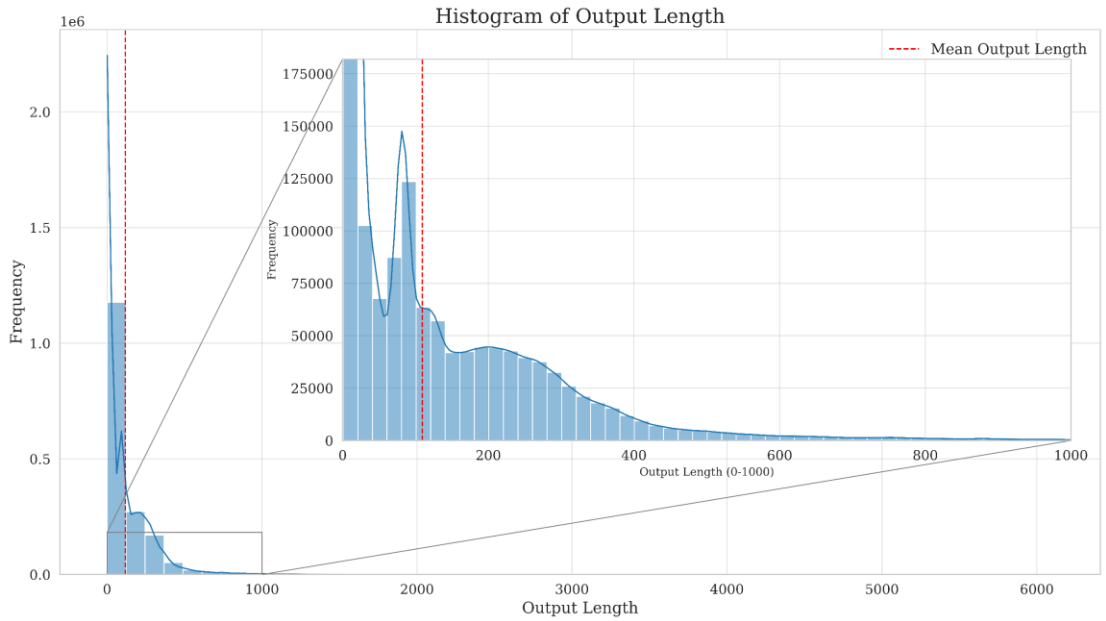

Figure3. Histogram of Output Length

Figure 3 illustrates the character count (output length) of the output text in the dataset, which exhibits a wide distribution range, covering both short and long texts. The distribution is concentrated in the 0 to 1000 character range. Short texts (texts with fewer characters) appear more frequently, and as the output length increases, the frequency decreases. Kernel Density Estimation (KDE), also known as Parzen's window[28], is one of the most renowned methods for estimating the underlying probability density function of a dataset. The KDE curve provides a smooth estimate of the distribution within this range, aiding in a more intuitive understanding of the text distribution pattern.

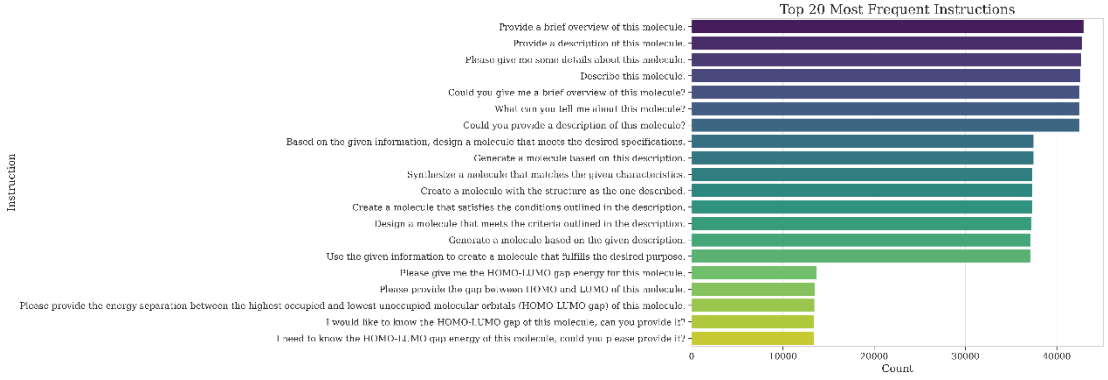

Figure4. Top 20 Most Frequent Instructions

The bar chart (Figure4) shows the frequency of the 20 most common instructions in the dataset for this study. Among these, "**Provide a brief overview of this**

molecule" and "**Provide a description of this molecule**" appear significantly more often than other instructions, indicating their prominent role in the dataset. Nonetheless, other types of instructions also appear, demonstrating the diversity of instruction types within the dataset.

Table 1 List of datasets used in our study.

Dataset	Url link	Data format
ESOL[43]	https://github.com/MasterAIEAM/Darwin/blob/main/dataset/ESOL/ESOL.json	Json
MoosaviCp[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/MoosaviCp/MoosaviCp.json	Json
MoosaviDiversity[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/MoosaviDiversity/MoosaviDiversity.json	Json
NagasawaOPV[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/NagasawaOPV/NagasawaOPV.json	Json
Chembl[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/chembl/chembl.json	Json
matbench_expt_gap[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/matbench_expt_gap/matbench_expt_gap.json	Json
matbench_glass[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/matbench_glass/matbench_glass.json	Json
matbench_is_metal[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/matbench_is_metal/matbench_is_metal.json	Json
matbench_steels[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/matbench_steels/matbench_steels.json	Json
Pei[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/Pei/pei.json	Json
waterStability[43]	https://github.com/MasterAI-EAM/Darwin/blob/main/dataset/	Json

	waterStability/waterStability.json	
description_guided_molecule_design[44]	https://huggingface.co/datasets/zjunlp/Mol-Instructions/tree/main/data	Json
forward_reaction_prediction[44]	https://huggingface.co/datasets/zjunlp/Mol-Instructions/tree/main/data	Json
molecular_description_generation[44]	https://huggingface.co/datasets/zjunlp/Mol-Instructions/tree/main/data	Json
reagent_prediction[44]	https://huggingface.co/datasets/zjunlp/Mol-Instructions/tree/main/data	Json
property_prediction[44]	https://huggingface.co/datasets/zjunlp/Mol-Instructions/tree/main/data	Json
Retrosynthesis[44]	https://huggingface.co/datasets/zjunlp/Mol-Instructions/tree/main/data	Json

3.2 Fine-tuning

In this work, we collected and curated 1,720,313 fine-tuning instructions from the field of chemical science. Using the unsloth[29] tool, we fine-tuned open-source large language models including llama-3-8B-Instruct-bnb-4bit,mistral-7B-instruct-v0.3-bnb-4bit,gemma-7B-bnb-4bit,gemma-2-9b-bnb-4bit,Phi-3-mini-4k-instruct, Mistral-Nemo-Instruct-2407-bnb-4bit and Llama-3.1-8B-Instruct-bnb-4bit.We employed the PEFT (Parameter-Efficient Fine-Tuning) method to apply LoRA (Low-Rank Adaptation) technique for fine-tuning the pre-trained models. The training parameters were configured using SFTTrainer and TrainingArguments. By combining quantization techniques, LoRA technology, and optimized training configurations, we aimed to enhance performance and optimize resource utilization. Table2 Parameter settings for the fine-tuning process for LLMs.

Table2. Fine-tuning Process Parameter Settings

Parameter	Value	Description

lora_alpha	16	LoRA alpha parameter
max_steps	60	Maximum training steps
learning_rate	2e-4	Learning rate
weight_decay	0.01	Weight decay parameter
seed	3407	Random seed

Fig5 represents the training loss curve during the training process of LLMs. In the initial phase of training, the loss value is relatively high because the model parameters have not yet been optimized, leading to a significant gap between the predicted results and the actual values. As the training progresses, the model gradually learns and continuously adjusts the parameters, making the predicted results increasingly closer to the actual values. Consequently, the error decreases, and the loss value gradually declines and tends to stabilize.

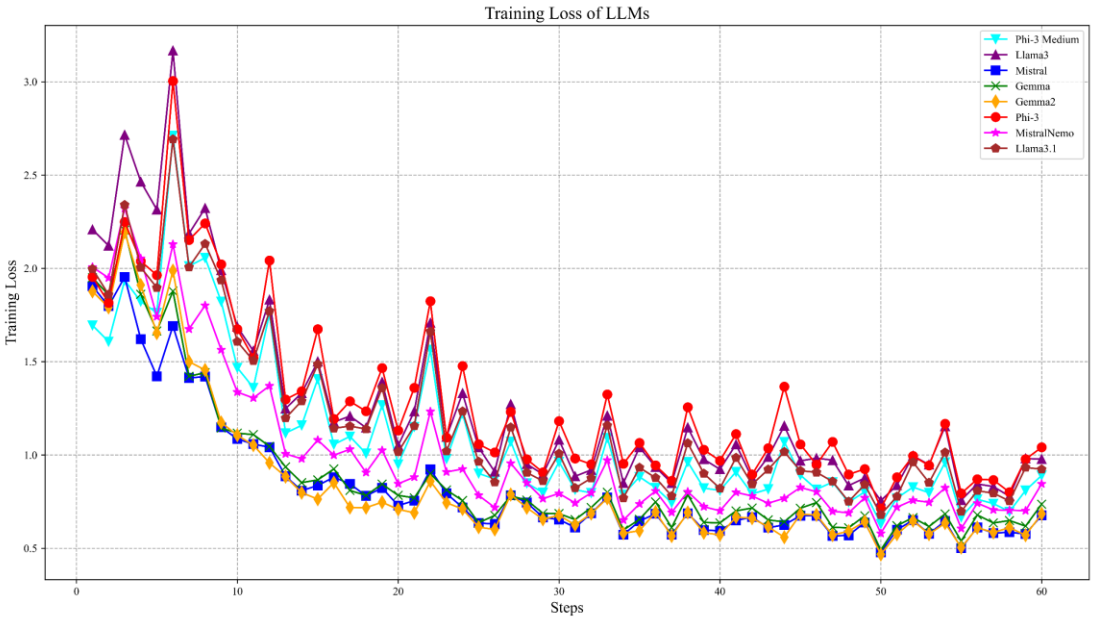

Fig 5.Training Loss of LLMs

3.3 Deployment of LLMs (Large Language Models)

After the fine-tuning step in Section 2.2 of the large language model, we employed Ollama for the local deployment and testing of fine-tuned LLMs. Model parameters were set using the Modelfile configuration file. Specifically, the model's

temperature was set to 0.8, and the context window size was configured to 8192 tokens. Additionally, three stop markers were defined to control the boundaries of the generated text. The detailed configuration is shown in the Fig6. After fine-tuning, the four large language models were deployed on a local computer for testing. The four fine-tuned large language models(Llama3-8B,Phi-3-mini,Gemma-7B,Mistral-7B) were deployed on a local computer with an Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz (up to 2.11 GHz) and an NVIDIA GeForce MX250 GPU for testing. The two fine-tuned models are tested using Google Colab, with Gemma2-9B tested on a T4 GPU, Phi-3Medium tested on an L4 GPU,Llama3.1-8B tested on Colab CPU and Mistral Nemo tested on L4 GPU.

```
FROM ./Name.gguf

TEMPLATE """{{- if .System }}
<|system|>
{{ .System }}
{{- end }}
<|user|>
{{ .Prompt }}
<|assistant|>
"""

SYSTEM """You are a helpful, smart,
kind, and efficient AI assistant.Your
name is (Set according to your
preferences). You always fulfill the
user's requests to the best of your
ability."""

PARAMETER temperature 0.8
PARAMETER num_ctx 8192
PARAMETER stop "<|system|>"
PARAMETER stop "<|user|>"
PARAMETER stop "<|assistant|>"
```

Fig6. Model parameter specific settings

3.4 Methods for Evaluating the Quality of LLM Responses

Based on previous research, evaluation after fine-tuning large language models is crucial, as it serves as a key tool for identifying current system limitations and informing the design of more powerful models[30]. Therefore, in this work, to assess the performance of different large models after fine-tuning, 100 questions were randomly selected from the dataset for model testing. To evaluate the performance of

different models after fine-tuning more objectively, this study specifically designed OptimizedModelEvaluator, an automatic scoring program to evaluate the performance of different models.

Different scoring criteria were designed for different questions. Additionally, the evaluator considered some special cases in the field of chemical science, assigning higher weights to key words such as 'reaction', 'mechanism', 'synthesis', and 'catalyst'. It also recognizes specific chemical terms (e.g., 'alkane', 'alkene', 'alkyne'), considers conversions between different units when making numerical comparisons (such as kJ to kcal), and applies special processing for questions involving specific concepts like LUMO, HOMO, and orbital energies (comparing the signs (positive or negative) of the extracted answer value and the correct answer value; LUMO and HOMO energies are typically negative, so the correctness of the sign is important). For questions involving MOFs, it pays special attention to key concepts such as 'linker', 'node', and 'topology'.

The system employs various methods to evaluate the quality of answers. For numerical problems, it calculates relative errors and assigns corresponding scores. It uses Levenshtein distance[31] or simple word set intersections to compute the similarity between answers and standard solutions. BLEU scores[32] and ROUGE scores[33] are used to assess the quality of generated text and summaries, respectively. The Flesch[34] Reading Ease Index is utilized to evaluate text readability. In addition to these methods, the system also incorporates evaluation criteria such as keyword relevance, coherence, conciseness, factual accuracy, and creativity.

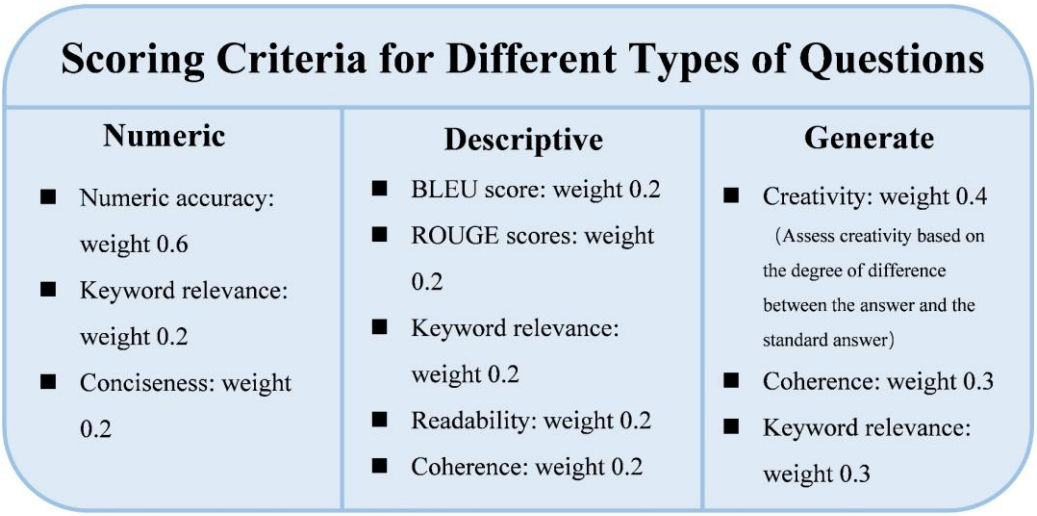

Fig7. Scoring Criteria for Different Types of Questions

Through these detailed settings, the evaluator can better assess the model's understanding of concepts related to molecular orbital theory, rather than just simple numerical matching. This enables a comprehensive evaluation of AI models' performance in answering chemistry-related questions, covering multiple dimensions including accuracy, relevance, readability, and creativity. Figure 8 illustrates the scoring process. (See supporting information for details).

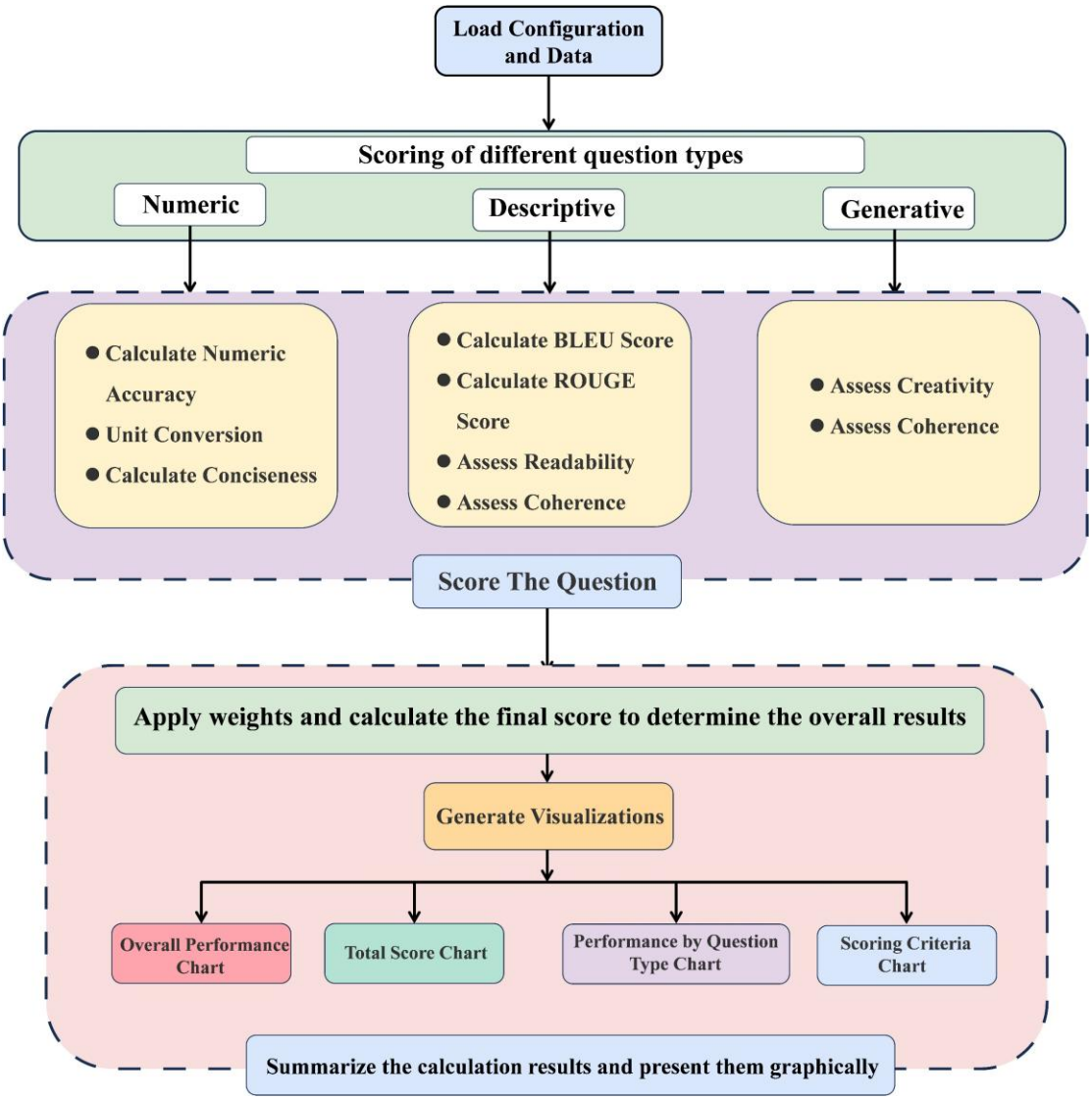

Fig8. Automatic Grading Program Process

3.5 LLMs Fine-Tuning Test Results and Discussion

This study conducts a comprehensive evaluation of six fine-tuned large language models: Llama3-8B, Mistral-7B, Phi-3 Mini, Gemma-7B, Gemma2-9B,Phi-3 Medium,Llama3.1 and MistralNemo. Through testing across multiple dimensions, we aim to gain a deep understanding of the performance differences of these models under various tasks and criteria, providing insights for model selection and future optimization directions. Using the automated scoring program introduced in Section 2.4, the fine-tuned models were evaluated with four main metrics: overall score, average performance, multi-dimensional criteria evaluation, and question type classification assessment. Each model was fine-tuned using the same strategy and tested on the same test set (details in the supporting information), ensuring the

comparability of the results.

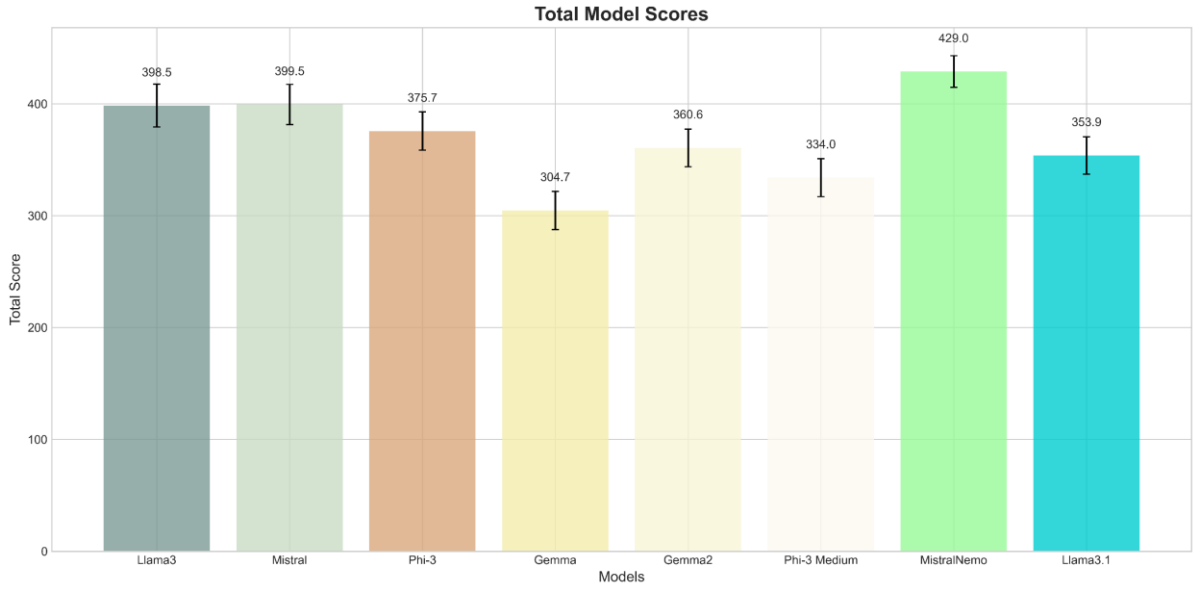

Fig9. Total scores for each model

The overall model scores (Figure9) show the total scores of eight models, with MistralNemo performing the best, scoring 429.0. Llama3 and Mistral follow closely behind, scoring 398.5 and 399.5 respectively, with both performing very similarly. Phi-3 follows with a score of 375.7. Notably, Gemma2 (360.6 points) shows significant improvement compared to its predecessor Gemma (304.7 points). The iteration from Mistral to MistralNemo also demonstrates the effectiveness of model iterations. However, Phi-3 Medium scored lower than Phi-3 Mini, possibly due to the increased number of parameters making model optimization more challenging, requiring more complex training strategies and computational resources. Additionally, Llama3.1 scored lower than Llama3, indicating that not all model iterations contribute to improved performance after fine-tuning.

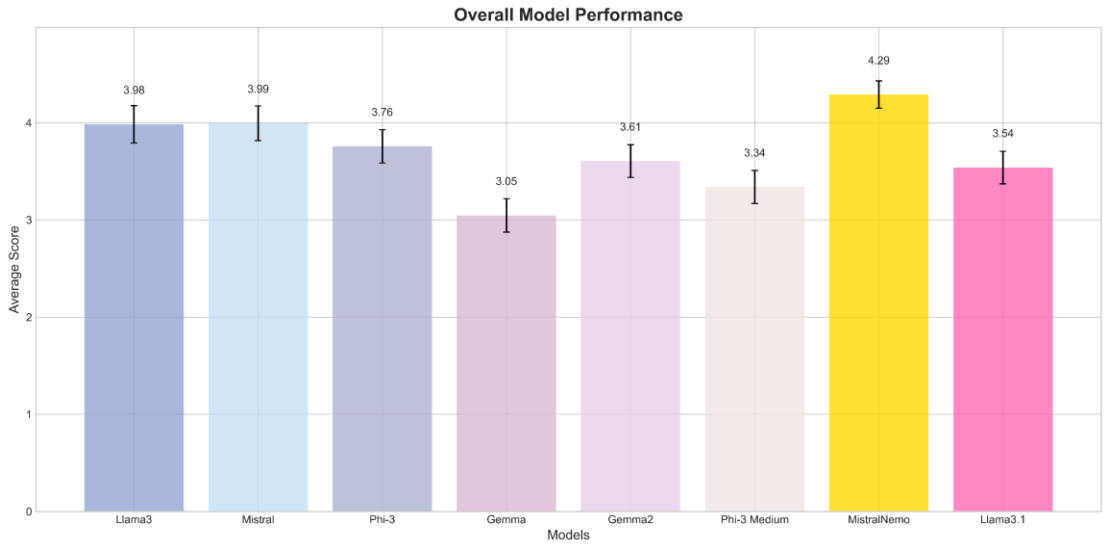

Fig10. Average scores for each model

Overall Model Performance (Figure10): Figure 10 presents the same ranking trend using a normalized 0-5 scale. This normalization allows for a more intuitive comparison of relative performance differences between models. The 0-5 scale is closer to common rating systems, making performance evaluation more intuitive and relative performance clearer. On the normalized scale, differences after the decimal point become more meaningful, making subtle performance changes more apparent while maintaining the overall structure and relationships of the data.

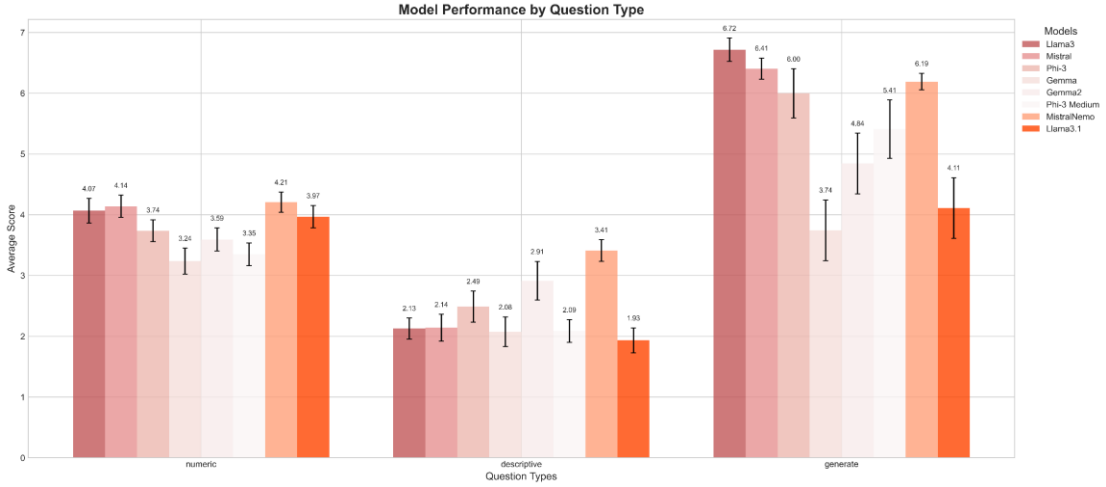

Fig11. Model Performance by Question Type

Model Performance Across Different Question Types (Figure11) categorizes questions into numerical, descriptive, and generative types, providing insights into model performance across different task natures. All models perform best on generative questions, with scores ranging from 3.74 to 6.72. This result aligns with

the high scores in creativity and coherence in Figure 11, further confirming the strong capabilities of large language models in open-ended generation tasks. Descriptive questions are the most challenging for all models, with scores ranging from 1.93 to 3.41, indicating room for improvement in precise description and information extraction. Performance on numerical questions falls between the other two types (3.24 to 4.21).

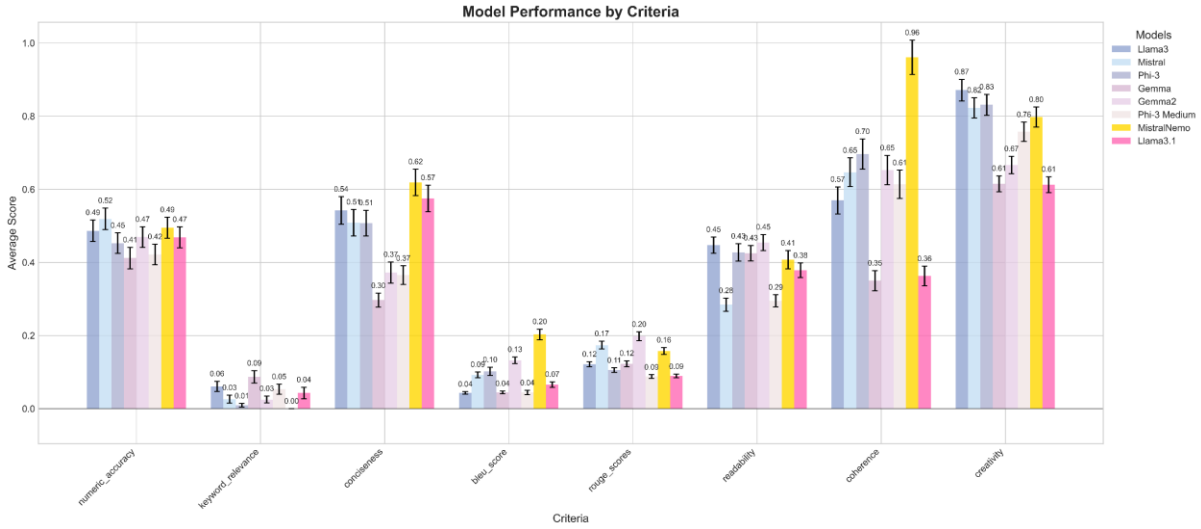

Fig12.Model Performance by Criteria

Model Performance Across Different Criteria (Figure12) provides a fine-grained analysis of model performance across eight key criteria (numerical accuracy, keyword relevance, conciseness, task score, response range, problem-solving ability, coherence, and creativity). Models excel in creativity and coherence, generally scoring above 0.6, reflecting the common strengths of large language models in generating fluent and creative text. Keyword relevance and task score are common challenges for all models, with scores generally below 0.2. This suggests that even after fine-tuning, models still have room for improvement in accurately grasping task requirements and key information. Mistral stands out in numerical accuracy, surpassing other models, reflecting its optimization effect on specific tasks. MistralNemo maintains a lead in most criteria, showcasing its comprehensive performance advantage.

Research findings reveal the significant impact of model iterations on performance improvement, particularly evident in the evolution from Gemma-7B to Gemma2-9B[35] and from Mistral-7B to Mistral-Nemo. However, the iteration from

Llama3-8B to Llama3.1-8B failed to achieve the expected performance leap, possibly due to different iteration priorities[36]. Notably, all tested models face common challenges, especially in keyword relevance and task scoring, highlighting the necessity of introducing additional technologies to address these shortcomings.

Nevertheless, the outstanding performance of these models in creative and generative tasks continues to demonstrate the inherent advantages of large language models in these domains. Test results indicate that fine-tuned large language models can meet researchers' needs to some extent, but still have many limitations, including the inability to update data in real-time, lack of online search capabilities, poor compatibility with specific domains, insufficient response accuracy, and limitations in decision-making for single large models.

Given these limitations exhibited by fine-tuned large language models, this study developed an artificial intelligence assistant for the chemical domain with autonomous evolution capabilities. This system cleverly integrates multi-agent architecture, Retrieval-Augmented Generation (RAG) technology, online search functionality, dynamic learning and evolution mechanisms, as well as a user-friendly interactive interface, aiming to comprehensively address the aforementioned shortcomings and provide researchers with a more intelligent, precise, and practical auxiliary tool.

4. Self-Evolving AI Agents for Chemistry

This work builds upon the fine-tuning of the aforementioned large language models to design an AI assistant platform specifically tailored for the field of chemistry. The platform integrates multi-agent systems, retrieval-augmented generation, real-time web search, and chemical structure visualization. The system incorporates AI agents with diverse professional backgrounds (such as laboratory directors, senior chemists, safety officers, etc.), simulating a virtual chemistry research team environment. These agents can collaborate, continuously learn and evolve to provide researchers with comprehensive and professional support in chemical knowledge, experimental design suggestions, safety guidance, and data

analysis. Additionally, the system has the capability to convert chemical structure formulas (SMILES) into visualized images, greatly enhancing the efficiency and intuitiveness of chemical research, education, and team collaboration. The system primarily consists of the following components: Multi-agent system, Retrieval-augmented generation (RAG), Real-time web search, Chemical structure visualization, Agent evolution system and User-friendly interface design.

4.1 Multi-Agent System

This system is the core architecture of the project, simulating a real chemical team. The system contains five specialized agents, each with a specific role and expertise, together forming a comprehensive and efficient virtual chemical research team. The Lab_Director is responsible for overall task allocation and research direction guidance, ensuring the team's research direction aligns with the overall goals and coordinating work between agents. The Senior_Chemist provides in-depth chemical knowledge and solutions to complex problems, possessing rich chemical theory and practical experience to handle challenging chemical issues and propose innovative research ideas. The Lab_Manager is responsible for experiment planning and resource management, ensuring the feasibility of experimental plans, managing laboratory resources, optimizing experimental processes, and improving research efficiency. The Safety_Officer ensures all discussions and suggestions comply with safety standards, focusing on experimental safety, reviewing potential risks of all experimental protocols, and providing safety operation guidance. The Analytical_Chemist focuses on data analysis and instrument use, responsible for interpreting experimental data, providing instrument operation advice, and ensuring data accuracy and reliability. This design allows each agent to have its specific area of expertise, providing in-depth professional knowledge. Agents can complement each other to solve complex problems collaboratively. For example, when the Senior_Chemist proposes an experimental protocol, the Safety_Officer reviews its safety, while the Lab_Manager considers its feasibility. This multi-perspective analysis allows agents with different backgrounds to analyze problems from various angles, providing comprehensive insights. The structure simulates the team dynamics

of a real chemistry research group, closely mimicking real team decision-making processes. Each agent in the system is based on a large language model but has specific system prompts to define its role and expertise, and different language models can be substituted to meet the needs of different tasks. AutoGen is used to manage interactions and dialogue flow between agents, adopting a round-robin approach to select speakers, ensuring each agent has the opportunity to contribute. The above multi-agent design allows the system to analyze and solve chemical problems from multiple perspectives, providing comprehensive insights (This research uses the fine-tuned and performance-tested MistralNemo as a main model for this system, and all fine-tuned large language models involved in this research have been uploaded to Hugging Face and can be set to call different large language models according to different needs KANGYONGMA/Chemistry).

4.2 Retrieval-Augmented Generation (RAG)

RAG is a core functionality of the system, extending the knowledge base of agents by integrating preloaded chemical literature and experimental data. The RAG workflow includes document loading, text splitting, vector embedding, vector storage, similarity search, context enhancement, and answer generation. This process is implemented using the langchain library and RetrievalQA chain, significantly improving the accuracy and relevance of answers while reducing the possibility of AI generating false information. RAG technology enables agents to provide answers based on the latest chemical research, cite relevant literature to support views, and associate user queries with existing knowledge bases, thereby greatly enhancing the system's ability to handle complex chemical problems and provide more precise and relevant information.

4.3 Real-time Web Search

Another important feature of the system is the ability to perform real-time web searches by integrating the Tavily search API[40] to supplement the preloaded knowledge base. The workflow of this feature includes query analysis, API calls, result processing, and information integration. The system uses the requests[41] library to send API requests and implements error handling and retry mechanisms to ensure

stability. This feature allows agents to access the latest chemical research and discoveries, supplement information that may be missing from the preloaded database, and significantly improve the system's ability to answer current affairs questions. By combining preloaded data and real-time search, the system can provide users with comprehensive, up-to-date, and accurate chemical information, excelling particularly in handling emerging research, latest discoveries, or real-time data-related issues.

4.4 Chemical Structure Visualization

This feature greatly enhances the system's interactivity and intuitiveness when discussing chemical structures by converting SMILES[5] strings into 2D molecular structure images. The entire process involves SMILES parsing, molecular object creation, 2D coordinate generation, image rendering, and encoding, ultimately displaying on the Web interface. This functionality not only enhances the visual understanding of chemical concepts and improves the efficiency of discussing complex molecular structures but also makes the system more suitable for chemical education and research applications. Its implementation mainly relies on the RDKit[42] library for molecular manipulation and image generation, integrating it into the message processing flow to achieve automatic detection and conversion of SMILES strings, thereby providing chemistry researchers with a more intuitive and effective chemical structure interaction experience.

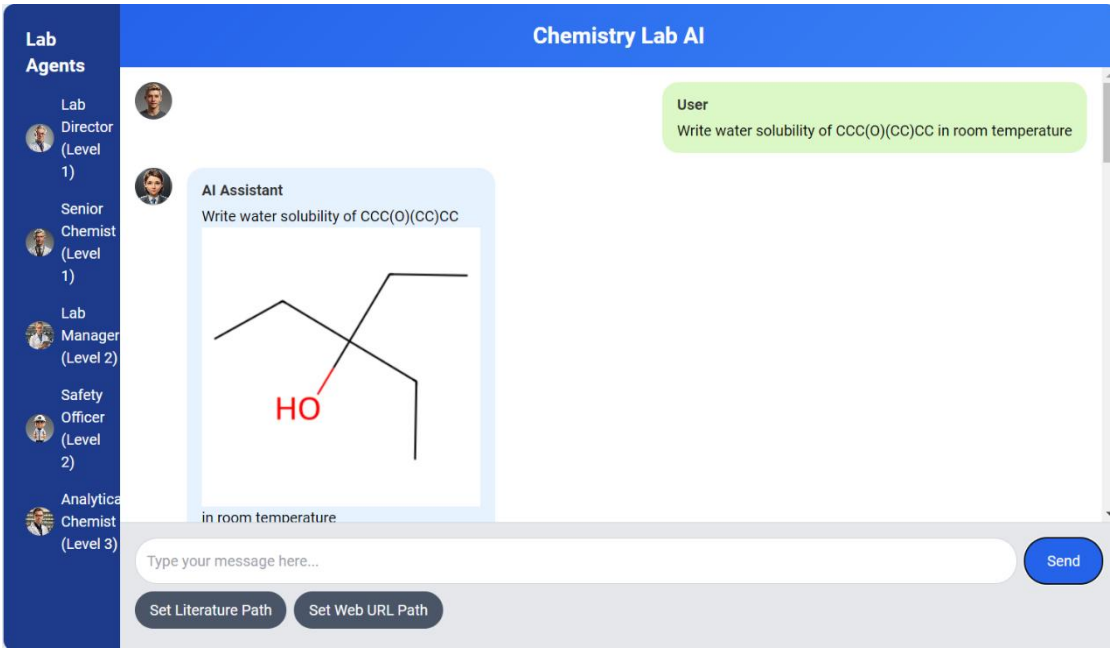

Fig14. Dialogue Interface SMILES Visualization

4.5 Agent Evolution System

Agent evolution is an innovative feature that allows agents to improve their performance and knowledge base over time. The system includes four main components: knowledge acquisition, skill development, performance evaluation, and adaptive adjustment. Through mechanisms such as knowledge base expansion, skill tree updates, feedback learning, and cross-learning, agents can learn new chemical concepts, acquire new problem-solving abilities, and adjust behaviors based on user feedback. The system uses dynamic data structures to store knowledge and skills, implements a scoring system to quantify performance, and adopts probability models to simulate the evolution process. This dynamic learning mechanism enables agents to continuously improve, adapt to user needs, and simulate human learning and professional development processes. Over time, agents continuously enhance their capabilities, providing increasingly relevant and useful information to users, thereby significantly improving the overall performance and user experience of the system. This system core is composed of two main classes: ChemistryAgent and ChemistryLab, implementing functions such as knowledge accumulation, skill acquisition, performance evaluation, and group evolution. The ChemistryAgent class stores knowledge and skills through the knowledge_base and skills attributes, constantly expanding its capabilities using the learn() and acquire_skill() methods. The performance evaluation mechanism records recent performance through performance_history, and the evaluate_performance() method assesses performance based on user feedback. The evolution mechanism is triggered by the evolve() method, determining whether to enhance or improve skills based on average performance. The improve() and refine_skills() methods are responsible for acquiring new skills and optimizing existing skills, respectively. The system can also identify areas for improvement by analyzing interaction history. At the group level, the ChemistryLab class implements knowledge sharing among agents and multi-round evolution simulation. This design allows the system to continuously adjust and optimize based on actual interactions and feedback, continuously improving its professional

capabilities and interaction quality in the field of chemistry, forming a dynamically adaptive and self-improving intelligent ecosystem.

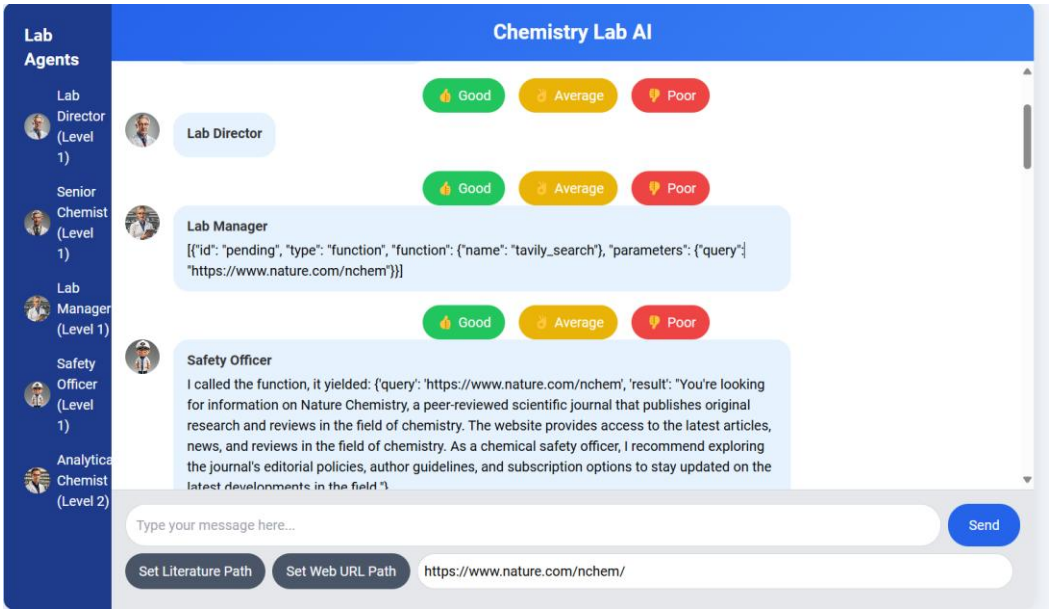

Fig.15 User Feedback and Intelligent Agent Evolution Interface

4.6 User-friendly interface design

The project includes an intuitive web interface that can display real-time conversations between agents, agent status, and feedback mechanisms, providing a better interactive experience.

Function Expansion:

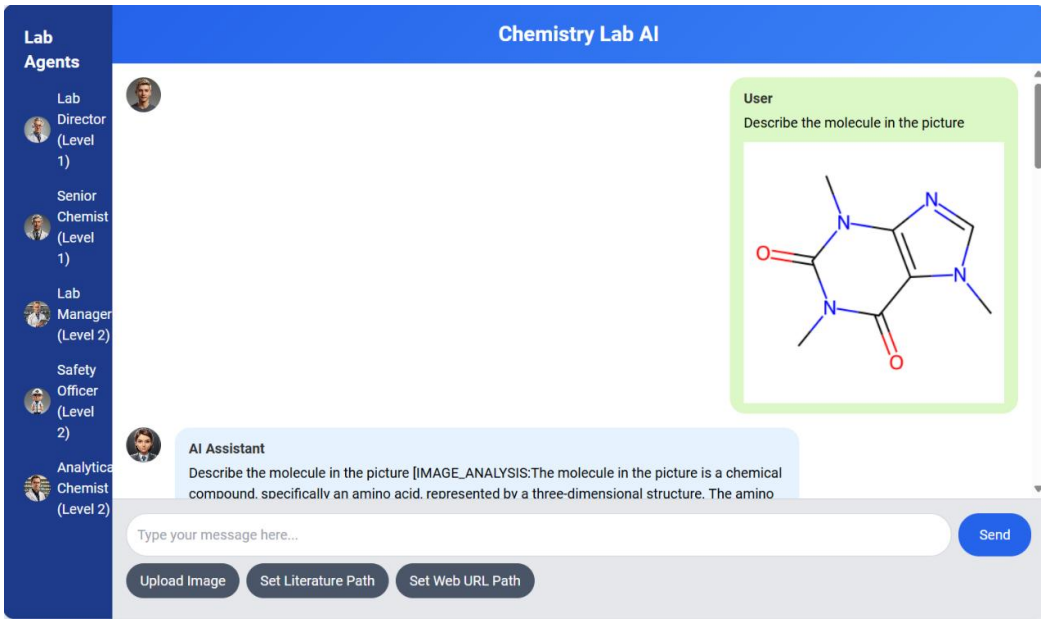

Fig.16 Functionality Expansion—Multimodal Models

4.7 Functionality Expansion

During the system design phase, the team fully considered the potential impact of model update and iteration, and therefore reserved corresponding upgrade and development space. Figure 16 demonstrates the image recognition capabilities after the integration of multi-modal large models, which provides an important foundation for expanding more functionalities in the future.

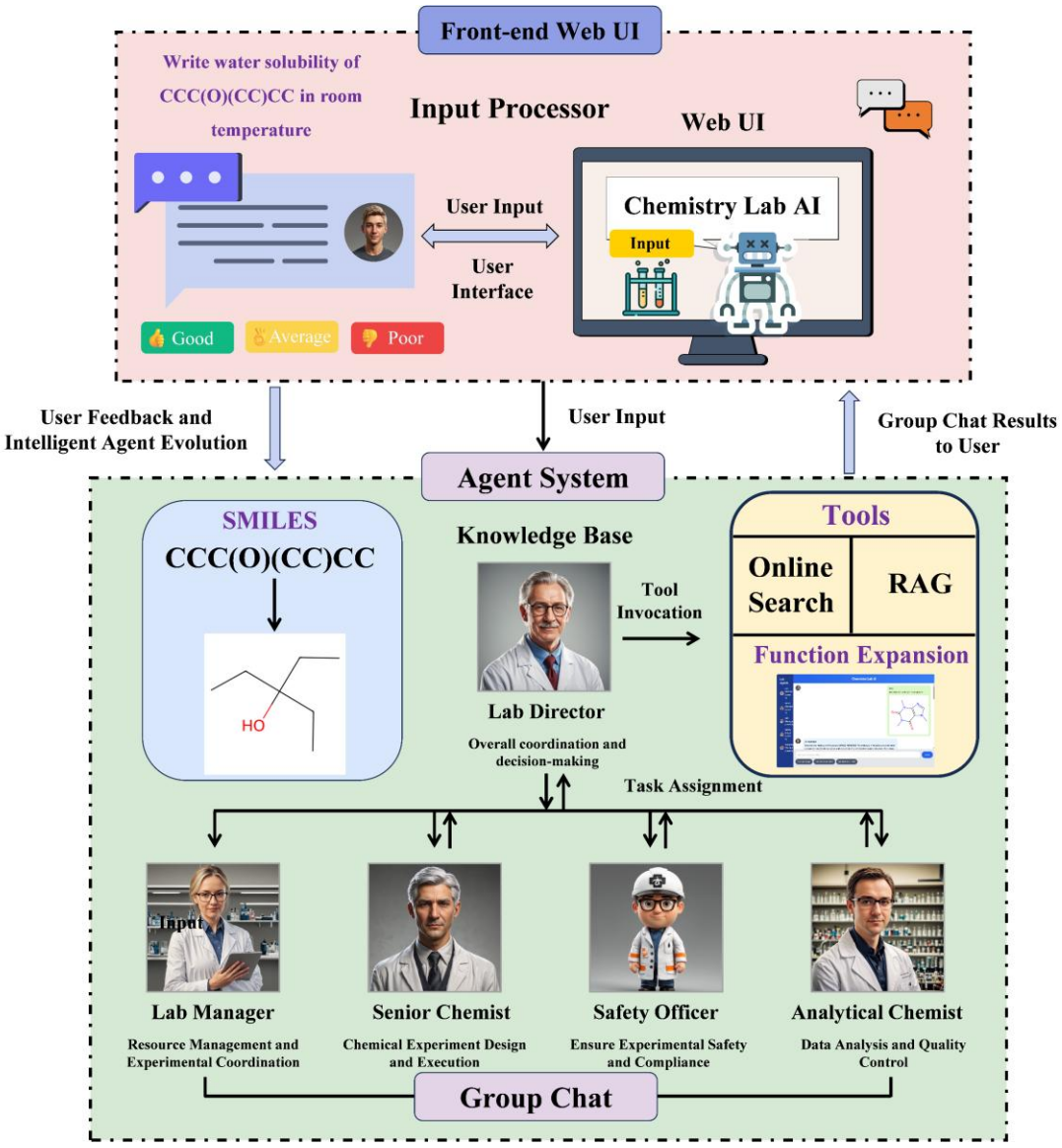

Fig 17. The Structure of Self-Evolving AI Agents for Chemistry System

The system's design fully considers the rapid iteration characteristics of large language models, implementing a flexible mechanism to call upon different advanced models. The system deeply integrates specialized functions in the field of chemistry, such as molecular visualization and SMILES string processing, precisely meeting the needs of chemical research.

The core advantage of the system lies in its autonomous evolution capability. Through knowledge accumulation, skill acquisition, performance evaluation, and group collaboration, it can continuously optimize its professional capabilities and interaction quality. This dynamic adaptive feature breaks through the static limitations of traditional AI systems, providing intelligent and efficient support for solving complex chemical problems.

Conclusion

This study utilized 1,720,313 instruction data points from the field of chemical science to fine-tune 8 mainstream open-source large language models, including Llama3-8B, Mistral-7B, Phi-3 Mini, Gemma-7B, Gemma2-9B, Phi-3 Medium, Llama3.1, and MistralNemo. Through an automatic scoring program specifically designed to evaluate the quality of responses from large language models in the chemistry domain, the MistralNemo model demonstrated the most outstanding performance, achieving a total score of 429 points, surpassing other models.Based on these results, an innovative chemical intelligent assistant system was designed. This system employs the fine-tuned Mistral Nemo model as its primary model and can call upon different large models according to task requirements. Furthermore, the system deeply integrates professional knowledge and requirements from the chemistry field, featuring specialized functionalities such as molecular visualization, SMILES string processing, and chemical literature retrieval.Benefiting from knowledge accumulation, skill acquisition, performance evaluation, and collaborative mechanisms, the system can continuously optimize its professional capabilities and interaction quality. This allows the system to learn and grow continuously, breaking through the inherent static limitations of traditional AI systems and opening up new possibilities for the application of artificial intelligence in the field of chemistry.

Acknowledgements

This work involves the following AI technologies: Llama3-8B, Mistral-7B, Phi-3 Mini, Gemma-7B, Gemma2-9B, Phi-3 Medium, Llama3.1, and MistralNemo. The

aforementioned open-source large language models were used for fine-tuning tests in this work. Tavily Search AI was used for online searches, and sentence-transformers/all-mpnet-base-v2 was used for RAG (Retrieval-Augmented Generation). Additionally, Claude 3.5 Sonnet was used to address code issues encountered in this research, assist in developing the Web UI interface, optimize the multi-agent framework, and expand multi-agent tools. The Agent avatar in this work was generated by Stable Diffusion 3.The manuscript was polished using Claude 3.5 Sonnet and ChatGPT-4o. We are grateful for the assistance of these AI technologies in completing this work.

Author contributions

Kangyong Ma was responsible for the conception and design of this study. He conducted the data analysis and interpretation. He wrote the original draft of the manuscript and created the visualizations.

Conflict of interest

The authors have no conflicts of interest to declare.

Data availability

The code for this work is available at https://github.com/KangyongMa/GVIM, while the data and models can be found at https://huggingface.co/KANGYONGMA.

Funding

This work has no financial support.

References

[1] Pavlick E. Symbols and grounding in large language models[J]. Philos Trans A Math Phys Eng Sci, 2023, 381(2251): 20220041.

[2] Chen W, Yan-Yi L, Tie-Zheng G, et al. Systems engineering issues for industry applications of large language model[J]. Applied Soft Computing, 2024, 151: 111165.

[3] Xiao H, Zhou F, Liu X, et al. A comprehensive survey of large language models and multimodal large language models in medicine[J]. arXiv.org, 2024. arXiv:2405.08603.

[4] Han Z, Gao C, Liu J, et al. Parameter-efficient fine-tuning for large models: a comprehensive survey[J]. arXiv.org, 2024. arXiv:2403.14608.

[5] Livne M, Miftahutdinov Z, Tutubalina E, et al. Nach0: multimodal natural and chemical languages foundation model[J]. Chemical Science, 2024.

[6] Ouyang L, Wu J, Xu J, et al. Training language models to follow instructions with human feedback[J]. arXiv.org, 2022. arXiv:2203.02155.

[7] Wei J, Bosma M, Zhao V Y, et al. Finetuned language models are zero-shot learners[J]. arXiv.org, 2022. arXiv:2109.01652.

[8] Sanh V, Webson A, Raffel C, et al. Multitask prompted training enables zero-shot task generalization[J]. arXiv.org, 2022. arXiv:2110.08207.

[9] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning[J]. arXiv.org, 2021. arXiv:2104.08691.

[10] Nori H, King N, Mckinney S M, et al. Capabilities of gpt-4 on medical challenge problems[J]. arXiv.org, 2023. arXiv:2303.13375.

[11] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. arXiv.org, 2021. arXiv:2005.11401.

[12] Guu K, Lee K, Tung Z, et al. Realm: retrieval-augmented language model pre-training[J]. arXiv.org, 2020. arXiv:2002.08909.

[13] Vladimir Karpukhin B O G S, Ledell Wu S E D C. Dense passage retrieval for open-domain question answering[J]. arXiv.org, 2020. arXiv:2004.04906.

[14] Petroni F, Lewis P, Piktus A, et al. How context affects language models' factual

predictions[J]. arXiv.org, 2020. https://openreview.net/forum?id=025X0zPfn

[15] Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: a survey[J]. arXiv.org, 2024. arXiv:2312.10997.

[16] Xi Z, Chen W, Guo X, et al. The rise and potential of large language model based agents: a survey[J]. arXiv.org, 2023. arXiv:2309.07864.

[17] Wooldridge, M. J., N. R. Jennings. Intelligent agents: theory and practice. Knowl. Eng. Rev., 10(2):115–152, 1995.

[18] Shoham, Y. Agent oriented programming. In M. Masuch, L. Pólos, eds., Knowledge Representation and Reasoning Under Uncertainty, Logic at Work [International Conference Logic at Work, Amsterdam, The Netherlands, December 17-19, 1992], vol. 808 of Lecture Notes in Computer Science, pages 123–129. Springer, 1992.

[19] Hutter, M. Universal artificial intelligence: Sequential decisions based on algorithmic probability. Springer Science & Business Media, 2004.

[20] Wu Q, Bansal G, Zhang J, et al. Autogen: enabling next-gen llm applications via multi-agent conversation[J]. arXiv.org, 2023. arXiv:2308.08155

[21] Tian L, He Z, Jiao W, et al. Encouraging divergent thinking in large language models through multi-agent debate[J]. arXiv.org, 2023. arXiv:2305.19118

[22] Du Y, Li S, Torralba A, et al. Improving factuality and reasoning in language models through multiagent debate[J]. arXiv.org, 2023. arXiv:2305.14325

[23] Wu Y, Jia F, Zhang S, et al. An empirical study on challenging math problem solving with gpt-4[J]. arXiv.org, 2023. arXiv:2306.01337

[24] Shimin Li T S Q C. Agent alignment in evolving social norms[J]. arXiv.org, 2024. arXiv:2401.04620

[25] Cheng Qian S L Y Q, Sun M. Investigate-consolidate-exploit: a general strategy for inter-task agent self-evolution[J]. arXiv.org, 2024. arXiv:2401.13996

[26] Merz K M, Wei G, Zhu F. Editorial: harnessing the power of large language model-based chatbots for scientific discovery[J]. Journal of Chemical Information and Modeling, 2023, 63(17): 5395.

[27] Stokel-Walker, C., & Van Noorden, R. (2023). What ChatGPT and generative AI

mean for science. Nature, 614(7947), 214-216.
[28] Chen, Y. C. (2017). A tutorial on kernel density estimation and recent advances. Biostatistics & Epidemiology, 1(1), 161–187.
[29] Daniel Han and Michael Han. 2024. Unsloth, Unsloth AI. Company name: Unsloth AI Package/product name: Unsloth.
[30] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. ACM Transactions on Intelligent Systems and Technology, 15(3), Article 39, 1-45. https://doi.org/10.1145/3641289
[31] Coates P, Breitinger F. Identifying document similarity using a fast estimation of the levenshtein distance based on compression and signatures[J]. arXiv.org, 2023.
[32] Ehud Reiter; A Structured Review of the Validity of BLEU. Computational Linguistics 2018; 44 (3): 393–401. doi: https://doi.org/10.1162/coli_a_00322
[33] Zhang M, Li C, Wan M, et al. Rouge-sem: better evaluation of summarization using rouge combined with semantics[J]. Expert Systems with Applications, 2024, 237: 121364.
[34] Hershenhouse J S, Mokhtar D, Eppler M B, et al. Accuracy, readability, and understandability of large language models for prostate cancer information to the public[J]. Prostate Cancer Prostatic Dis, 2024.
[35] Gemma Team. (2024). Gemma 2: Improving Open Language Models at a Practical Size. In International Conference on Learning Representations (ICLR). OpenReview.net.
[36] Llama Team, AI @ Meta. (2024). The Llama 3 Herd of Models. Retrieved from https://llama.meta.com/
[37] Wu Q, Bansal G, Zhang J, et al. Autogen: enabling next-gen llm applications via multi-agent conversation[J]. arXiv.org, 2023.
[38] crewAI from crewAI - Platform for Multi AI Agents Systems
[39] H Chase. 2023. LangChain LLM App Development Framework. Retrieved July 10, 2023 from https://langchain.com/
[40] Tavily Search API from https://tavily.com/

[41] Kenneth Reitz. Requests: HTTP for Humans. Available at: https://requests.readthedocs.io/en/.

[42] Scalfani, V.F., Patel, V.D. & Fernandez, A.M. Visualizing chemical space networks with RDKit and NetworkX. J Cheminform 14, 87 (2022).

[43] Xie T, Wan Y, Huang W, et al. DARWIN Series: Domain Specific Large Language Models for Natural Science[J]. arXiv.org, 2023. arXiv:2308.13565

[44] Fang Y, Liang X, Zhang N, et al. Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models[C]. ICLR. OpenReview.net, 2024.

[45] Jablonka K M, Schwaller P, Ortega-Guerrero A, et al. Leveraging large language models for predictive chemistry [J]. Nature Machine Intelligence, 2024, 6: 161–169. https://doi.org/10.1038/s42256-023-00788-1.

[46] Xie Z, Evangelopoulos X, Omar ÖH, Troisi A, Cooper AI, Chen L. Fine-tuning GPT-3 for machine learning electronic and functional properties of organic molecules. Chem. Sci. 2024, 15, 500-510. DOI: 10.1039/d3sc04610a.

[47] Zhong, S., & Guan, X. (2023). Developing Quantitative Structure–Activity Relationship (QSAR) Models for Water Contaminants' Activities/Properties by Fine-Tuning GPT-3 Models. *Environmental Science & Technology Letters*.

[48] Kim, S., Jung, Y., & Schrier, J. (2024). Large Language Models for Inorganic Synthesis Predictions. *Journal of the American Chemical Society*, 146(29), 19654-19659. doi: 10.1021/jacs.4c05840

[49] Bran, A. M., Cox, S., Schilter, O., Baldassari, C., White, A. D., & Schwaller, P. (2024). Augmenting large language models with chemistry tools. Nature Machine Intelligence, 6, 525–535. https://doi.org/10.1038/s42256-024-00832-8

[50] Boiko, D.A., MacKnight, R., Kline, B. *et al.* Autonomous chemical research with large language models. *Nature* **624**, 570–578 (2023). https://doi.org/10.1038/s41586-023-06792-0

[51] McNaughton, A. D., Ramalaxmi, G., Kruel, A., Knutson, C. R., Varikoti, R. A., & Kumar, N. (2024). CACTUS: Chemistry Agent Connecting Tool-Usage to Science. Preprint. https://arxiv.org/abs/2405.00972

717

718

719

720

721