Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2025, **4**, 2827

DropMicroFluidAgents (DMFAs): autonomous droplet microfluidic research framework through large language model agents

Dinh-Nguyen Nguyen, Raymond Kai-Yu Tong and Ngoc-Duy Dinh *

Large language models (LLMs) have gained significant attention in recent years due to their impressive capabilities across various tasks, from natural language understanding to generation. Applying LLMs within specific domains requires substantial adaptation to account for the unique terminologies, nuances, and context-specific challenges inherent to those areas. Here, we introduce DropMicroFluidAgents (DMFAs) employing LLM agents to perform two key functions: (1) delivering focused guidance, answers, and suggestions specific to droplet microfluidics and (2) generating machine learning models to optimise and automate the design of droplet microfluidic devices, including the creation of code-based computer-aided design (CAD) scripts to enable rapid and precise design execution. To assess the accuracy of DMFAs in question–answering tasks, we compiled a dataset of questions with corresponding ground-truth answers and established an evaluation criterion. Experimental evaluations demonstrated that integrating DMFAs with the LLAMA3.1 model yielded the *highest accuracy of 76.15%*, underscoring the significant performance enhancement provided by agent integration. This effect was particularly pronounced when DMFAs were paired with the GEMMA2 model, resulting in *a 34.47% improvement in accuracy* compared to the standalone GEMMA2 configuration. For evaluating the performance of DMFAs in design automation, we utilized an existing dataset on flow-focusing droplet microfluidics. The resulting machine learning model demonstrated *a coefficient of determination of approximately 0.96*. To enhance usability, we developed a streamlined graphical user interface (GUI) that offers an intuitive and effective means for users to interact with the system. This study demonstrates the effective use of LLM agents in droplet microfluidics research as powerful tools for automating workflows, synthesising knowledge, optimising designs, and interacting with external systems, bringing a significant transformation to the field of digital discovery. DMFAs is capable of transforming them into closed-loop digital discovery platforms that encompass literature synthesis, hypothesis generation, autonomous design, execution in self-driving laboratories, analysis of results, and the generation of new hypotheses. These capabilities enable their application across education and industrial support, driving greater efficiency in scientific discovery and innovation.

Received 11th July 2025
Accepted 16th August 2025

DOI: 10.1039/d5dd00306g

rsc.li/digitaldiscovery

1. Introduction

Droplet microfluidics is a cutting-edge technology that leverages microchannel networks to manipulate discrete droplets as independent microreactors. This technology minimises sample consumption, reduces waste, and allows for high-throughput processing, making it indispensable for applications where scalability and precision are critical.[1–9] Droplet microfluidics, leveraging its foundational strengths, has driven innovations across various disciplines within the chemical and biological sciences, including advancements in next-generation sequencing,[10–12] single-cell RNA sequencing,[13,14] single cell secretion analysis,[15,16] drug screening,[17] droplet digital PCR,[18] and liquid biopsies diagnostics.[19] However, designing droplet microfluidics devices is complex and often requires iterative trial-and-error processes.[20–24] To address the complexity of device design, machine learning (ML) has emerged as a promising tool for automating the optimisation of droplet-based microfluidic systems.[25–32] However, implementing ML-based design automation requires expertise in both microfluidics and machine learning to optimise algorithms and interpret outcomes effectively. Additionally, considerable time is needed to comprehend and utilise the extensive body of prior knowledge in droplet microfluidics and machine learning documented in the scientific literature, which is critical during the initial stages of experimentation.

LLMs are advanced artificial intelligence (AI) systems capable of understanding and generating human-like text by

Department of Biomedical Engineering, The Chinese University of Hong Kong, Room 208, Ho Sin Hang Engineering Building (SHB), Shatin, N. T., Hong Kong, China. E-mail: ngocduydinh@cuhk.edu.hk

processing vast amounts of data.[33] LLMs have demonstrated broad applicability across various disciplines, with notable contributions in specialised areas such as chemistry,[34–38] biology,[39–43] biomedical research,[44–47] materials science,[48–54] and medicine[55–60] offering significant benefits in automating and enhancing research processes. Furthermore, LLMs have played a pivotal role in advancing scientific discovery and fostering technological innovation.[61–64] However, the inherent limitations of LLMs, such as hallucination, bias, and incomplete factual accuracy, necessitate careful oversight and validation of their outputs.[65,66] To overcome some of these limitations, Retrieval-Augmented Generation (RAG) frameworks combine LLMs with external knowledge retrieval systems, enabling more accurate and contextually relevant outputs.[67] RAG models retrieve pertinent information from structured databases or scientific repositories, ensuring the generated content aligns with verified knowledge. However, basic RAG implementations face scalability challenges, limited retrieval accuracy, and dependence on the quality of the external sources.[68] LLM agents are advanced AI systems that go beyond generating text by acting as autonomous agents capable of planning, reasoning, and executing tasks. Unlike traditional LLMs, which passively respond to inputs, agentic LLMs can interact with external tools, APIs, and databases to achieve specific goals. These agents operate autonomously, performing complex, multi-step tasks such as iterative querying, hypothesis testing, and experimental design.[69–74] These agents have found applications in autonomous scientific discovery[75] and medical research.[76–78] However, their deployment in droplet microfluidics remains underexplored, representing a significant opportunity for innovation.

In this study, we introduce DropMicroFluidAgents (DMFAs), a novel multi-agent-based framework designed to perform diverse tasks and make decisions autonomously, guided by its programming and the data it analyses. DMFAs comprises two components, the Scientific Mentor and the Automation Designer, both constructed using cutting-edge LLMs, as shown in Fig. 1. The Scientific Mentor delivers customised guidance and recommendations aimed at enhancing theoretical understanding and reducing the trial-and-error costs associated with experimental workflows in droplet microfluidics. Notably, the Automation Designer is capable of developing a machine learning model to optimise and automate the design of droplet microfluidic devices, as well as providing code-based CAD scripts for drawing creation. This study emphasises the transformative potential of LLMs in advancing droplet microfluidics development, setting the stage for the rapid acceleration of scientific discovery and technological innovation through AI.

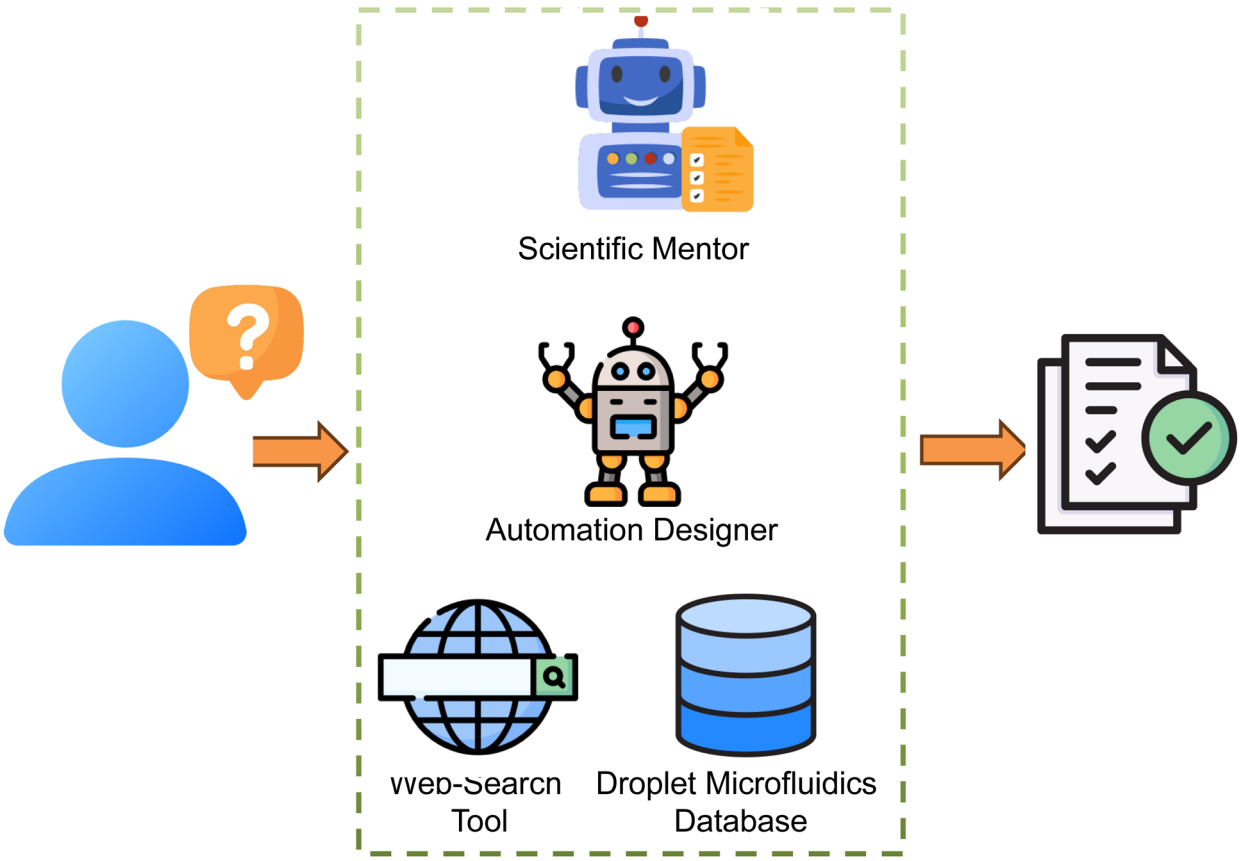


Fig. 1 The overall framework of DMFAs. The Scientific Mentor is tasked with providing guidance to users by leveraging LLM agents and the specialized knowledge database, supplemented by the web search tool when required. Meanwhile, the Automation Designer, also leveraging LLM agents, facilitates the automation of the design process for users.

2. Methods

2.1 Large language model selection

The selection of LLMs is a critical decision in the design and optimisation of intelligent agents. For this study, three models available through the Ollama framework including LLAMA 3.1 (8B parameters),[79] MISTRAL (7B parameters),[80] and GEMMA2 (9B parameters)[81] were selected for their distinct strengths, each tailored to address the specific requirements of the task as illustrated in Table 1. LLAMA 3.1 is renowned for its general-purpose capabilities and flexibility, making it particularly effective for a wide range of query types. It provides a solid foundation for handling both structured and unstructured inputs, adapting well to various contexts. The strength of LLAMA 3.1 lies in its ability to generate coherent responses across diverse topics. MISTRAL, on the other hand, excels in its ability to perform sophisticated contextual reasoning. Its architecture is specifically optimised to handle complex queries that require deep comprehension and multi-step inference. The strength of MISTRAL lies in its capacity to maintain context across extended conversations or intricate problem-solving tasks, which makes it particularly valuable for applications requiring high levels of logical consistency and nuanced understanding. The advanced attention mechanisms of this model allow it to consider multiple factors in parallel, offering a higher degree of precision when managing intricate relationships between data points or abstract concepts. GEMMA2 was selected for its specialised proficiency in domain-specific applications, particularly within technical fields such as microfluidics and machine learning. Its design emphasises high efficiency and accuracy when dealing with structured knowledge bases, which makes it ideal for tasks that involve retrieving and synthesizing specialized information. The ability of GEMMA2 to process and filter relevant domain-specific content quickly allows it to provide highly accurate, context-sensitive answers within narrow scopes, ensuring that the system can deliver expert-level insights in specialized areas. Building agents on these models involves not only leveraging their inherent strengths but also addressing critical factors such as scalability, latency, and fine-tuning for task-specific requirements. The integration of these LLMs ensures a balanced approach to accuracy, efficiency, and contextual relevance. Furthermore, their combined deployment allows for redundancy and cross-validation of outputs, enhancing the overall reliability and robustness of the system. This deliberate selection and integration underscore the importance of aligning model capabilities with the nuanced demands of the application domain.

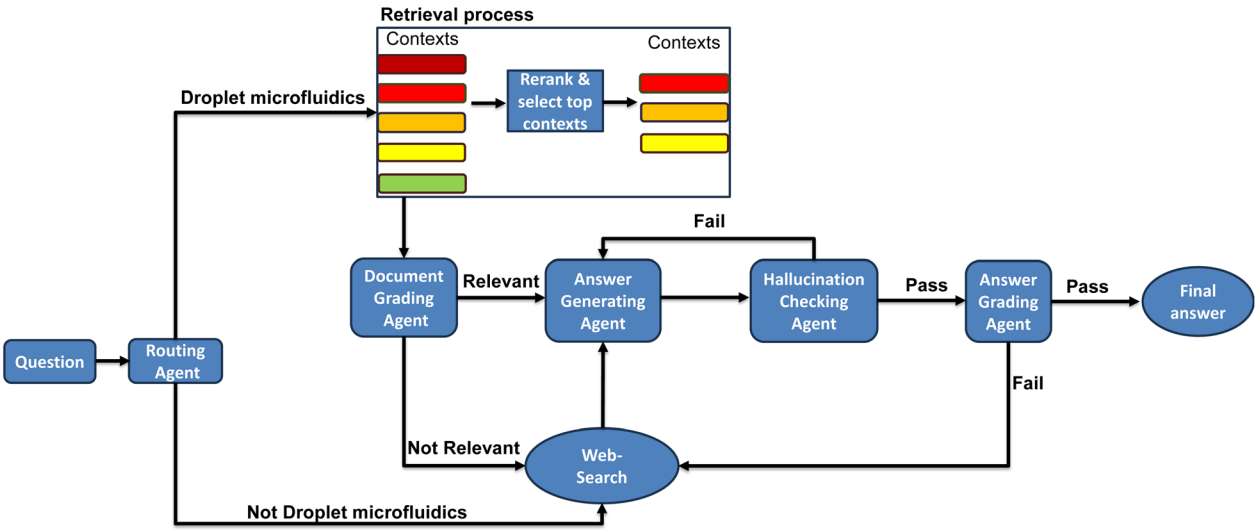
2.2 Scientific Mentor construction

2.2.1 Flowchart illustrating the operational principle of the Scientific Mentor. The Scientific Mentor represents a comprehensive and adaptive pipeline for a robust question–answering system, integrating RAG with mechanisms for relevance grading, hallucination detection, and iterative refinement, as illustrated in Fig. 2a. The system initiates with a routing agent that categorizes the question of a user as either pertaining to the microfluidic knowledge base or necessitating external context. For questions linked to the microfluidic knowledge base, the retrieval process employs methods such as embedding similarity or semantic search to identify the most relevant documents from the base. These retrieved documents are subsequently reranked to select the top three. These selected documents are then evaluated by a document grading agent, which assesses their relevance based on their alignment with the query. If irrelevant documents are identified, the pipeline redirects to the web-search tool for broader context or supplementary data, ensuring comprehensive coverage. The answer-generating agent is tasked with synthesizing responses based on the retrieved and evaluated documents. A critical safeguard is the hallucination checking agent, where the generated answer is scrutinized for unsupported or fabricated claims, leveraging fact-checking techniques. If hallucinations are detected, the system iterates by re-fetching or supplementing data, potentially from the answer-generating agent, ensuring the refinement of the response. The final step involves answer validation, where the answer grading agent verifies if the generated response adequately addresses the original question. If the response is unsatisfactory, the process loops back to the web search to gather additional information to refine the answer further. This architecture balances internal knowledge utilisation with external search capabilities, emphasising reliability and adaptability at the cost of increased computational overhead and potential delays.

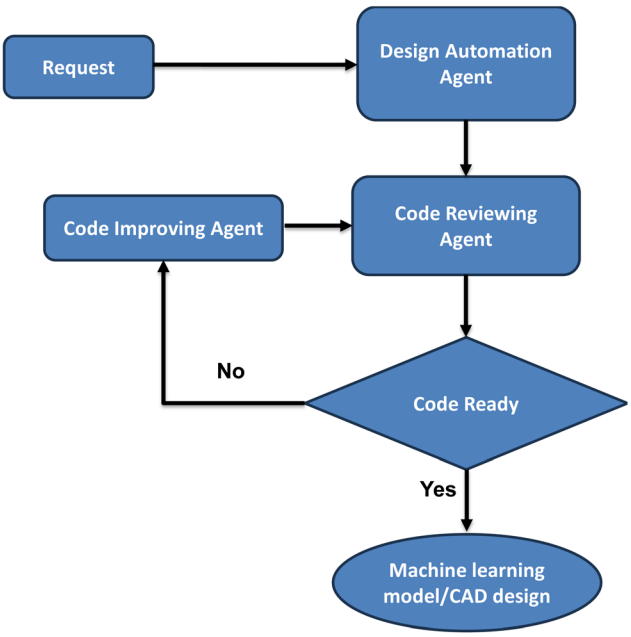
The Scientific Mentor is implemented sequentially using LLAMA 3.1, MISTRAL, and GEMMA2, represented as LLAMA-based Scientific Mentor, MISTRAL-based Scientific Mentor, and GEMMA-based Scientific Mentor, respectively. The accuracy

Table 1 LLMs used

Model	Parameter count	Architecture highlights	Training data composition	Deployment precision	Quantization applied?	Open-sourced?	Provider
Gemma 2:9B	9 billion	Cross-attention tuned for structured data	Scientific literature, technical manuals, knowledge graphs	FP16	Yes	Yes	Google
Mistral 7B	7 billion	Sliding-window attention; latency-optimised	Filtered datasets focusing on reasoning, logic, and academic texts	FP16	Yes	Yes	Mistral AI
Llama 3.1:8B	8 billion	Decoder-only transformer with rotary embeddings	Broad, diverse corpus (web, code, multilingual text)	FP16	Yes	Yes	Meta



(a) Scientific Mentor

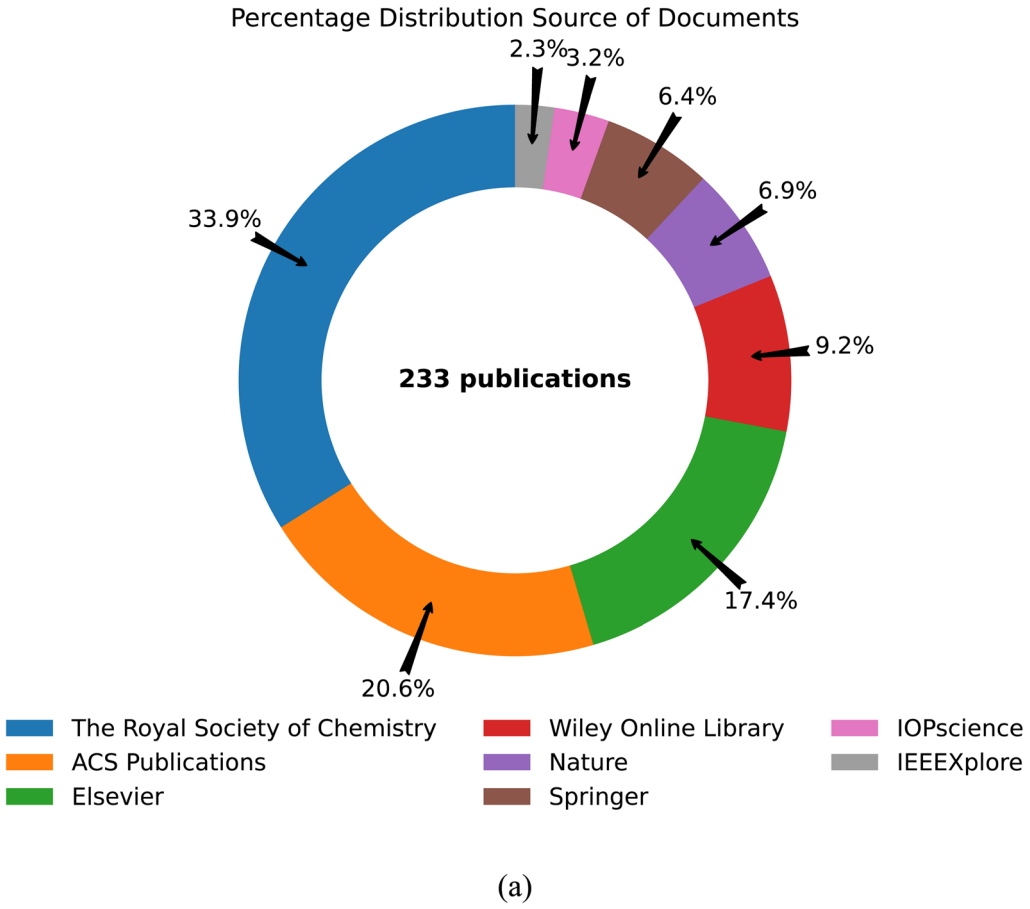


(b) Automation Designer

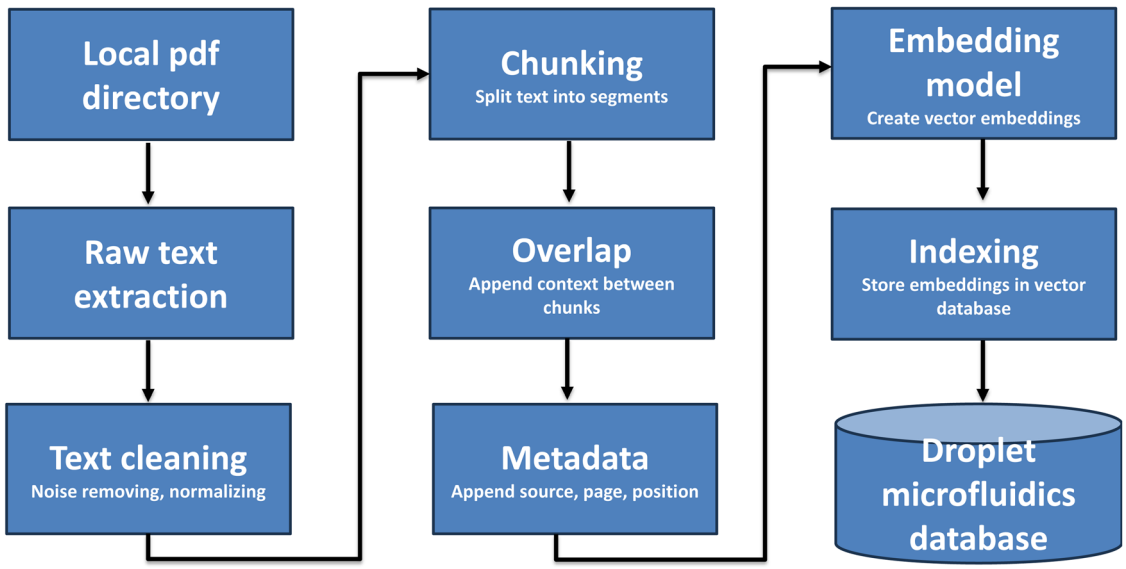
Fig. 2 Details of DMFAs. (a) Illustrates how the Scientific Mentor delivers the final answer to the user by utilizing multiple LLM agents to minimize the limitations of LLMs and improve the accuracy of the response. (b) Illustrates how the Automation Designer provides the machine learning model and CAD design to the user by utilizing three LLM agents to ensure an error-free result.

of the LLMs-based Scientific Mentor is compared to the accuracy of the standalone models, LLAMA 3.1, MISTRAL, and GEMMA2, in the question–answering (QA) task. Their performance is subsequently evaluated in the 'Results and discussion' section.

2.2.2 Generation of droplet microfluidics database and ground truth question–answer set. LLMs, as large general-purpose models trained on extensive text corpora, can produce inaccurate or nonspecific responses to highly specialised, domain-specific queries. However, incorporating domain-specific knowledge bases and supplementary tools can enhance their accuracy and enable them to provide more relevant and precise guidance.[67] To assess the efficacy of the Scientific Mentor in addressing question–answering challenges within droplet microfluidics, we constructed a droplet microfluidics database and a ground truth question–answer dataset based on 233 scientific papers sourced from a range of well-known publishers, as shown in Fig. 3a (SI, S1). The database



(a)



(b)

Fig. 3 (a) Percentage distribution source of documents (b) flowchart of the droplet microfluidics database.

has not been extensively expanded because our primary aim is to illustrate the operational effectiveness of the proposed framework rather than to create an exhaustive repository. However, the framework is designed to be extensible, allowing end-users to enrich the database with tailored data sources such as scientific publications, textbooks, and experimental reports.

This flexibility ensures that the framework can be adapted to specific domains and user needs. The ground truth question–answer dataset consists of 61 question–answer pairs encompassing a broad spectrum of topics related to microfluidics. These include fundamental concepts of microfluidics, principles of microfluidic device design, fabrication techniques, material considerations, applications in biology and medicine, chemical and material science applications, and prospects for the future of droplet microfluidics (SI, S2).

The construction of the droplet microfluidics database adheres to the process outlined in the flowchart presented in Fig. 3. This flowchart illustrates a pipeline for processing and storing text data extracted from documents, such as PDF files, enabling efficient semantic search and retrieval using the database. The process begins by dividing the content of the documents into manageable chunks, ensuring the text is appropriately segmented to preserve context and prevent information loss. These chunks are passed to an embedding model, a neural network typically trained on large datasets to convert textual data into high-dimensional embeddings, numerical representations that encode semantic meaning. The embeddings are then stored in the database for similarity search, enabling fast and accurate retrieval based on the semantic relationships between queries and stored data.

2.2.3 Evaluation method. The flowchart represents the workflow of a system designed to evaluate the accuracy of the Scientific Mentor by comparing its output against a predetermined ground truth, as shown in Fig. 4. The process begins with a question, which is directed to two components: the ground truth answer source and the Scientific Mentor. The ground truth answer represents the correct or expected response to the question, while the Scientific Mentor generates an automated response. Both the ground truth answer and the generated answer produced by the Scientific Mentor are then forwarded to an accuracy evaluation agent based on LLAMA 3.1, which serves as a comparator. This component systematically analyses the generated answer in relation to the ground truth answer to measure the accuracy performance of the Scientific Mentor using a 0–100 scoring scale, as shown in Table 2. The criteria emphasise factors such as relevance, clarity, coherence, conciseness, and depth of understanding in addressing the given question.

Lower scores (0–25) indicate responses that are irrelevant, nonsensical, or poorly structured, reflecting a lack of effort or focus on the question. Mid-range scores (26–55) represent partially correct responses with varying degrees of inaccuracies, verbosity, or insufficient depth, highlighting areas for improvement in precision and alignment with the query. Higher scores (56–85) denote responses that are largely relevant, clear, and accurate, albeit with minor issues such as unnecessary details or slight omissions. The top score range (86–100) reflects excellence, with responses being comprehensive, highly precise, and virtually flawless, showcasing exceptional clarity and direct alignment with the question. This tiered rubric provides a robust foundation for systematically evaluating and improving the performance of the Scientific Mentor in generating high-quality, contextually appropriate answers. Additionally, several supplementary metrics are used to evaluate the performance of the Scientific Mentor, including Bilingual Evaluation Understudy (BLEU), Recall-Oriented Understudy for

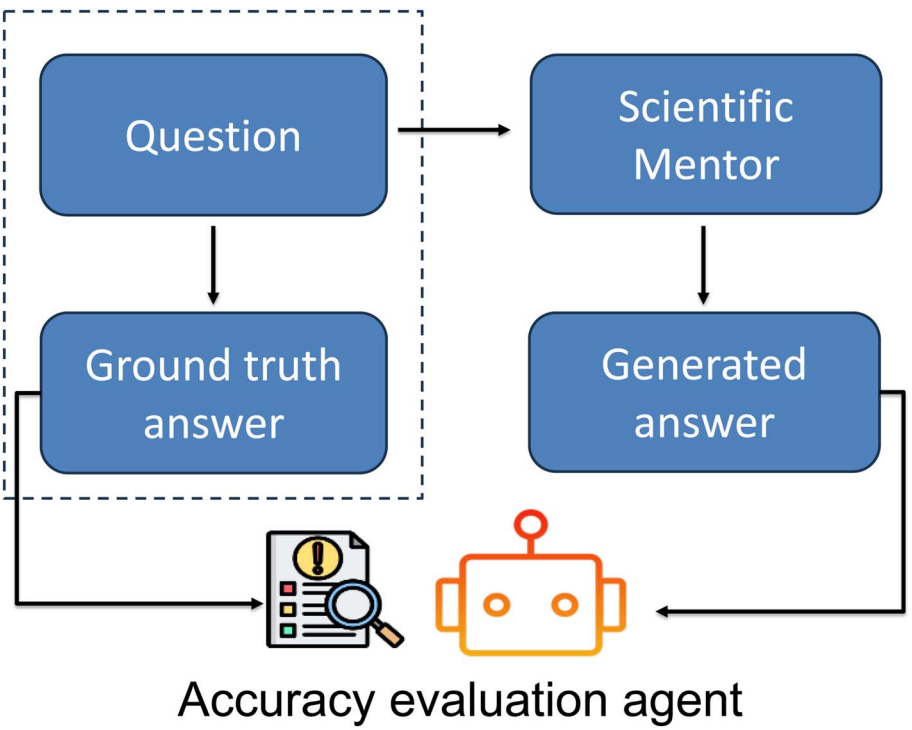


Fig. 4 Accuracy evaluation agent.

Table 2 Accuracy evaluation agent's criteria

Accuracy evaluation agent's criteria
Score 0–5: the response is irrelevant, nonsensical, or incoherent; no effort to address the question
Score 6–10: an attempt is made, but the response is entirely unrelated or meaningless
Score 11–15: contains vague or random terms but lacks clarity, relevance, and logic
Score 16–20: slight relevance, but the response is mostly incorrect, unclear, and fails to address the question
Score 21–25: marginal effort to address the question, but lacks clarity and coherence; mostly irrelevant
Score 26–30: some fragments of relevance, but the response is poorly structured and fails to convey useful information
Score 31–35: displays minimal understanding, with major inaccuracies and a lack of focus on the question
Score 36–40: partially aligned with the question but overly verbose, unclear, or dominated by factual errors
Score 41–45: demonstrates basic understanding but lacks conciseness, with significant errors or irrelevant details
Score 46–50: covers core aspects but is imprecise, verbose, or unclear; lacks depth or includes notable inaccuracies
Score 51–55: partially correct, with omissions or minor inaccuracies; somewhat clear and relevant but not concise
Score 56–60: mostly relevant and clear, but lacks focus or includes unnecessary details; broadly acceptable
Score 61–65: addresses the question clearly and concisely, with minor inaccuracies or slight verbosity
Score 66–70: a solid, clear, and mostly concise response that aligns with the question and ground truth
Score 71–75: comprehensive, clear, and precise, with only minor omissions or redundant elements
Score 76–80: nearly flawless response; highly clear, concise, and relevant, with only slight room for improvement
Score 81–85: a thorough and precise response; clear, concise, and directly addresses the question with full relevance
Score 86–90: excellent, highly clear, and succinct answer; covers all aspects comprehensively and precisely
Score 91–95: virtually flawless; exceptional clarity, conciseness, and relevance, with added depth where appropriate
Score 96–100: perfect response; unparalleled clarity, precision, and focus on the question, with no room for improvement

Gisting Evaluation (ROUGE), recall, F_1 score, and Metric for Evaluation of Translation with Explicit Ordering (METEOR), all of which are detailed in Table 3.

2.2.4 Investigation of how prompt design affects the accuracy of the Scientific Mentor. Prompt design is a fundamental aspect of optimising the performance of LLMs, as it significantly impacts the quality, relevance, and precision of their generated outputs. A well-crafted prompt not only provides clear instructions but also sets the context in which the model should operate, ensuring that the response aligns with the expectations of a user.[82–84] The prompt must define the task in a way that minimises ambiguity, especially for complex or domain-specific queries. Effective prompts often include specific cues, such as desired output formats or constraints, that guide the model's reasoning and response generation. For example, in technical fields, the prompt might include terminology and context that ensure the model can draw from relevant domain-specific knowledge rather than providing generalised or unrelated information. This level of clarity is crucial for ensuring that the LLM produces accurate and appropriate results, particularly when the task involves intricate subject matter or multi-step reasoning. In this context, the effectiveness of LLM agents in performing specific tasks is intrinsically tied to the precision of the prompts. A precise prompt enables the model to focus on the most relevant information, effectively guiding it through the process of generating the correct answer. This is especially important when the agent is tasked with complex problem-solving or when a high degree of contextual understanding is required. Inaccurately phrased or vague prompts can lead to a range of issues, including irrelevant answers, incomplete responses, or even misinterpretations of the query. For example, an ambiguous prompt may cause the model to generate a broad, generic response, rather than the highly specific answer necessary for technical or specialized tasks.

We systematically applied seven prompt engineering techniques to assess their impact on the accuracy of the LLAMA-based *Scientific Mentor*:

Table 3 Additional metrics for question–answering evaluation of the Scientific Mentor

Metric	Purpose
BLEU	Measures precision of n-gram overlaps between predictions and references
ROUGE-1	Measures overlap of unigrams (single words)
ROUGE-2	Measures overlap of bigrams (two-word sequences)
ROUGE-L	Measures the longest common subsequence, capturing fluency and coherence
Recall	Quantify the proportion of relevant instances that have been successfully retrieved, highlighting the completeness of the retrieval process
METEOR	Takes into account synonym matches and stemming, providing a more flexible approach
F_1 score	The harmonic mean of precision and recall at the token level between the prediction and ground truth

• Baseline – refers to our initial instruction comprising a direct task description.

• Zero shot – perform a task without any examples provided in the prompt.[82,85]

• Few shots – involves supplying the model with a small number of task-specific exemplars within the prompt to induce better performance.[86]

• Domain expert – ask the model to reflect the language and reasoning style characteristic of a field specialist.[87]

• Self-recitation – encourages the model to reiterate prior knowledge relevant to the task at hand before generating the final answer.[88]

• Chain of thought – guides the model to generate intermediate reasoning steps leading to a final answer.[89,90]

• Composite – combines two or more strategies, such as integrating few-shot examples with domain expert or chain of thought.[91,92]

Comprehensive descriptions of these prompts are available in the "modelfile.json" located at the linked GitHub repository.

2.2.5 Investigation of how embedding model affects to the accuracy of the Scientific Mentor. Embedding models in RAG systems transform textual content into high-dimensional vector representations for semantic search and retrieval. The choice of embedding model significantly affects retrieval accuracy, as it governs how well semantic relationships between queries and document chunks are captured.[93,94] Caspari *et al.*[95] evaluated the similarity of various embedding models within the context of RAG systems, highlighting the importance of selecting appropriate models to enhance retrieval performance. Their analysis revealed that certain open-source models exhibit high similarity to proprietary models, offering viable alternatives for RAG implementations.

In this study, we empirically examined the effect of the embedding model on the accuracy of the LLAMA-based *Scientific Mentor* when used with the baseline prompt. The evaluation was carried out across eight open-source embedding models available on **https://huggingface.co**, including *intfloat/e5-large-v2*,[96] *all-MiniLM-L6-v2*, *all-mpnet-base-v2*, *pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb*,[97] *BAAI/bge-base-en-v1.5*, *multi-qa-distilbert-cos-v1*, *msmarco-distilbert-dot-v5*, and *stsb-roberta-base-v2*.[98]

2.2.6 Investigation of how chunk size affects to the accuracy of the Scientific Mentor. In RAG systems, chunk size refers to the length of text segments into which source documents are divided for embedding and retrieval. The choice of chunk size significantly affects both retrieval relevance and generation quality.[99–102] Smaller chunks can improve retrieval precision by focusing on specific information but may fragment context and reduce semantic completeness. Larger chunks preserve more context but may dilute relevance and exceed token limitations.

In this study, we empirically assessed the effect of the chunk size hyperparameter on the accuracy of the LLAMA-based *Scientific Mentor* when used with our baseline prompt. The evaluation encompassed 12 distinct chunk size settings: 100, 200, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, and 5000 tokens.

2.2.7 Investigation of how chunk overlap affects to the accuracy of the Scientific Mentor. Chunk overlap involves the proportion of content shared between adjacent text segments during document preprocessing in RAG pipelines. Introducing overlap helps maintain contextual continuity across chunks, which is crucial when relevant information spans chunk boundaries. A moderate degree of overlap enhances retrieval coherence and reduces the risk of omitting critical details, thereby supporting more accurate downstream generation.[103] However, excessive overlap can lead to redundancy, increased computational overhead, and retrieval bias toward repetitive content.[103] Therefore, determining the optimal overlap percentage is essential for maintaining informational integrity and maximizing retrieval efficacy.

In this study, we empirically investigated the effect of the chunk overlap hyperparameter on the accuracy of the LLAMA-based *Scientific Mentor* under a baseline prompt setting. The analysis was conducted using a fixed chunk size of 2000 and spanned 11 predefined chunk overlap values: 100, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, and 1500 tokens.

2.2.8 Investigation of how temperature hyperparameter affects to the accuracy of the Scientific Mentor. Temperature is a pivotal hyperparameter in LLMs that modulates the randomness of token selection during text generation. It scales the logits before the softmax function, thereby influencing the probability distribution over the vocabulary. Lower temperatures yield more deterministic outputs, favoring high-probability tokens and enhancing coherence and factual accuracy.[104–107] Conversely, higher temperatures introduce greater variability, potentially fostering creativity but at the expense of reliability and precision.[104–107] However, the relationship between temperature and output quality is nuanced. A study by Peeperkorn *et al.*[108] found that while higher temperatures slightly increase novelty, they also lead to a moderate increase in incoherence, with no significant impact on cohesion or typicality. Furthermore, in clinical applications, Patel *et al.*[109] observed that varying the temperature had minimal effect on the accuracy of tasks such as mortality prediction and medical coding, challenging the assumption that lower temperatures are always preferable for factual tasks.

In this study, we empirically evaluated the impact of the temperature hyperparameter on the accuracy of the LLAMA-based *Scientific Mentor* when applied with our baseline prompt. The evaluation was conducted across 16 temperature settings, systematically varied from 0 to 1.5 in increments of 0.1.

2.2.9 Investigation of how top-*p* hyperparameter affects to the accuracy of the Scientific Mentor. Top-*p* sampling, also known as nucleus sampling, is a decoding strategy that selects tokens from the smallest possible set whose cumulative probability exceeds a threshold *p*. This method adapts to the shape of the probability distribution, allowing for dynamic adjustment of the candidate token set size. While top-*p* sampling can enhance the diversity of generated text, its impact on accuracy is complex. Holtzman *et al.*[110] demonstrated that top-*p* sampling mitigates the issue of repetitive and unnatural text often produced by deterministic decoding methods. However, at

Table 4 Prompt design for the Automation Designer

Agent name	Prompt's content
Design automation agent	For 'AutoLisp' case: you are an expert in writing code in AutoLisp. Generate the proper lines of code based on request. Only output the code and nothing else. Here is the request: {request} For 'Python' case: you are an expert in writing code in Python. Generate the proper lines of code based on request. Only output the code and nothing else. Here is the request {request}
Code reviewing agent	For 'AutoLisp' case: you are a code reviewer specialized in AutoLisp. You need to review the given code and potential bugs and point out issues as bullet list. Only output the bullet list and nothing else here is the code: {code} For 'Python' case: you are a code reviewer specialized in 'Python'. You need to review the given code following PEP8 guidelines and potential bugs and point out issues as bullet list. Only output the bullet list and nothing else. Here is the code: {code}
Code improving agent	For 'AutoLisp' case: you are an expert in improving AutoLisp code. Improve the given code given the following guidelines. Only output the improved code and nothing else. Here are the guidelines: {guidelines}. Here is the given code: {code} For 'Python' case: you are an expert in improving Python code. Improve the given code given the following guidelines. Only output the improved code and nothing else. Here are the guidelines: {guidelines}. Here is the given code: {code}

higher temperature settings, top-p sampling may struggle to balance coherence and creativity, leading to less reliable outputs. To address this, Nguyen *et al.*[111] proposed Min-P sampling, a dynamic truncation method that adjusts the sampling threshold based on the model's confidence, thereby improving coherence and quality even at high temperatures.

In this study, we conducted an empirical assessment of how the top-p sampling hyperparameter influences the accuracy of the LLAMA-based *Scientific Mentor* under our baseline prompting condition. The analysis spanned 11 systematically varied top-p values ranging from 0 to 1 in increments of 0.1.

2.2.10 Investigation of how top-k hyperparameter affects to the accuracy of the Scientific Mentor. Top-k sampling is a decoding technique where the model selects the next token from the k most probable candidates. This method introduces stochasticity while maintaining computational efficiency. Lower values of k tend to produce more focused and deterministic outputs, which can improve accuracy for factual tasks.[112] Higher values of k increase output variability, potentially enhancing creativity but also raising the risk of incoherence and semantic drift.[112]

In this study, we empirically examined the influence of the top-k sampling hyperparameter on the accuracy of the LLAMA-based *Scientific Mentor* under our baseline prompt configuration. The evaluation was performed across 16 top-k settings, systematically varied from 0 to 150 in increments of 10.

2.3 Autonomous Designer construction

2.3.1 Flowchart illustrating the operational principle of the Autonomous Designer. The flowchart depicts an agent-driven iterative workflow for the Autonomous Designer in droplet microfluidics, with autonomous agents managing each stage of the process, as illustrated in Fig. 2b. The workflow begins with the specific requirement that necessitates code creation. This input is passed to the design automation agent, which produces an initial version of the code based on predefined algorithms, templates, or learned patterns. The generated code is then evaluated by the code reviewing agent, which inspects the code for logical correctness, adherence to coding standards, and functional compliance. The process progresses to the code-ready stage, where it is assessed to ensure the code complies with all defined quality and functional requirements. If the code is deemed ready, the machine learning model or CAD design is produced. Otherwise, the code-improving agent intervenes to refine the code, addressing deficiencies through debugging, optimisation, or reworking based on feedback from the

Fig. 5 Human-centered validation of agent-generated CAD designs.

Table 5 An example demonstrates how the LLAMA-based Scientific Mentor answers a question and how the accuracy score for the response is determined

Question: how to generate microdroplets?
Ground-truth answer: microdroplet generation in microfluidics is achieved through two main methods: active and passive. Active methods utilize external forces like piezoelectric actuators to precisely control droplet size, formation rate, and spacing. These methods enable high uniformity in droplet generation, with deviations less than 0.3%. Passive methods rely on fluid properties and channel geometry, such as T-junctions or cross-flow configurations, to create droplets. These methods are influenced by factors like flow rate, viscosity, and interfacial tension, allowing for fine control over droplet size and uniformity. For example, PDMS-based microfluidic devices have been used for nanoparticle synthesis, where hydrodynamic coupling of nozzles generates droplet pairs for precise mixing. Both methods are valuable in applications like nanoparticle synthesis, offering high reproducibility and control over reaction conditions in biomedical and material science research LLAMA-based Scientific Mentor's answer: based on the provided context, there are several ways to generate microdroplets in microfluidic devices. According to document 3, which has a relevance score of 0.99394965, the most commonly used methods for microdroplet generation can be divided into two main categories: active and passive "Active Methods:" * A piezoelectric actuator-based droplet generator is described in document 2, which allows for independent control over droplet size, rate of formation, and distance between droplets. This device is able to generate microdroplets with high uniformity, deviating from the mean value by less than 0.3% "Passive Methods:" * The context mentions that PDMS-based microfluidic devices are used to generate microdroplets for bulk synthesis of nanoparticles (document 1). Specifically, a device designed by Frenz *et al.* enables the production of droplet pairs based on the hydrodynamic coupling of two separated nozzles * Another method mentioned is the use of thermally isolated zones in a droplet microreactor for the synthesis of TiO_2 nanoparticles Additionally, document 3 mentions that other methods have been explored to generate microdroplets, including * Cross-flow and T-junction configurations * Adjusting channel geometry, two-phase viscosity, flow rate, wettability, and interfacial tension These methods are used to achieve precise regulation over the size, shape, uniformity, and other factors of microdroplets Accuracy evaluation agent's feedback: the response provides a clear and detailed explanation of microdroplet generation methods, accurately categorizing them into active and passive techniques. It correctly describes the characteristics and applications of each method, including the use of piezoelectric actuators, PDMS-based devices, and thermally isolated zones. However, it does not fully address the question's request for a comprehensive overview, as it relies heavily on specific documents and figures mentioned in the provided context. SCORE: 81 Microfluidics expert rating: SCORE: 85 Standalone LLAMA's answer: microdroplets can be generated using various methods, including (1) Flow-focusing technique: this involves forcing a liquid through a narrow channel or nozzle, creating a stream of droplets (2) Pneumatic atomization: using compressed air to break up a liquid into small droplets (3) Ultrasonic nebulization: using high-frequency sound waves to create tiny droplets from a liquid (4) Electrostatic atomization: applying an electric charge to a liquid, causing it to break up into small droplets (5) Centrifugal atomization: spinning a liquid at high speed to create droplets (6) Laser-induced breakdown spectroscopy (LIBS): using a laser pulse to create microdroplets from a liquid (7) Microfluidic devices: specialized devices that use channels and valves to control the flow of liquids and generate microdroplets These methods can be used in various applications, such as drug delivery, chemical synthesis, and biological research Accuracy evaluation agent's feedback: the response provides a list of various methods to generate microdroplets but does not directly address the question's focus on the two main methods (active and passive) used in microfluidics. SCORE: 16 Microfluidics expert rating: SCORE: 10
Question: what key factors influence the design of flow-focusing droplet microfluidics optimized using machine learning?
Ground-truth answer: the design of flow-focusing droplet microfluidics, optimized using machine learning, depends on several critical factors. Channel geometry, including the shape and size, controls droplet generation by defining the break-off point where the dispersed phase meets the continuous phase. Flow rates of both phases affect droplet size and frequency, with higher continuous phase flow reducing droplet size. Viscosity influences flow behavior and shear forces, crucial for stable droplet formation. Shear forces between phases are controlled by adjusting channel geometry or flow rates, ensuring consistent droplet generation. Machine learning algorithms predict and optimize droplet diameter and generation rate by analyzing these design parameters, fluid properties, and large datasets, automating the design process for more efficient droplet production in life sciences applications LLAMA-based Scientific Mentor's answer: based on the provided context, the key factors that influence the design of flow-focusing droplet microfluidics optimized using machine learning are (1) "Channel geometry": the shape and size of the channel in which the fluids are flowing play a critical role in controlling droplet generation (2) "Flow rate": the rates at which the dispersed and continuous phases are forced through the channel affect the droplet diameter and generation rate (3) "Viscosity": the viscosity of the fluids also influences the droplet generation process, as it affects the flow behaviour and shear forces acting on the dispersed phase (4) "Shear forces": the shearing action between the continuous and dispersed phases is crucial in flow-focusing designs and can be controlled by adjusting the channel geometry or flow rates These factors are mentioned in document 2 (page_content = '49') as governing the generation of droplets in flow-focusing designs. Additionally, the use of machine learning algorithms to predict the performance of droplet generators based on these design parameters is discussed in document 1 (page_content = '22, 23…')

Table 5 (*Contd.*)

Question: what key factors influence the design of flow-focusing droplet microfluidics optimized using machine learning?
Accuracy evaluation agent's feedback: the response provides a clear and concise list of factors influencing flow-focusing droplet microfluidics optimized using machine learning. However, it lacks depth and detail compared to the ground truth answer, particularly in explaining how these factors interact with each other and with machine learning algorithms. Additionally, some information is presented as a simple statement without further explanation or context. SCORE: 51 Microfluidics expert rating: SCORE: 55
Standalone LLAMA's answer: the key factors that influence the design of flow-focusing droplet microfluidics optimized using machine learning include (1) Channel geometry and dimensions (2) Flow rates and ratios of the two immiscible fluids (3) Surface tension and interfacial properties (4) Viscosity and density of the fluids (5) Droplet size and polydispersity (6) Shear stress and pressure drop across the microchannel (7) Operating temperature and pressure
Accuracy evaluation agent's feedback: the answer provides a list of factors that influence the design of flow-focusing droplet microfluidics optimized using machine learning but lacks explanation and context. It appears to be a summary of key parameters without elaboration on how they interact or are used in the optimization process. SCORE: 21 Microfluidics expert rating: SCORE: 20

reviewing agent. This cycle continues iteratively until the code passes the readiness evaluation. The agent-driven approach ensures automation, consistency, and efficiency across all workflow stages. This design highlights the potential of agent-based systems to streamline iterative development while maintaining a focus on continuous improvement and quality assurance. LLAMA 3.1 was selected for implementing the Autonomous Designer. A detailed explanation of the Autonomous Designer's prompts is provided in Table 4.

2.3.2 Human-centered validation of agent-generated CAD designs in AutoCAD environments. In the proposed multi-agent framework, CAD design generation is facilitated through the automated production of AutoLISP scripts, which interface directly with the AutoCAD working environment to yield precise geometric renderings of the intended structures. Crucially, these outputs are not blindly adopted but undergo a critical phase of human verification and visual inspection within the AutoCAD interface prior to any downstream fabrication processes as shown in Fig. 5. This step ensures that the design not only adheres to the functional specifications but also conforms to practical constraints and user-defined criteria. The fidelity of AutoCAD's geometric visualization enables users to assess structural integrity, dimensional accuracy, and design feasibility in a high-fidelity virtual setting. Consequently, the integration of human oversight within an automated CAD workflow ensures both adaptability and reliability, bridging algorithmic design generation with the nuanced judgment of experienced practitioners.

3. Results and discussion

3.1 Performance of the Scientific Mentor

An illustrative example involving the evaluation of two questions demonstrates a systematic approach for assessing the quality of answers generated by the Scientific Mentor based on LLAMA and standalone LLAMA. This evaluation uses a predefined scoring framework, with the results detailed in Table 5, providing a comparative analysis of the performance of two models. The process begins with a specific question relating to microfluidics, which serves as the input for the system to generate a corresponding answer. This generated response is then compared against a ground truth answer, representing the expected or ideal response to the given question. The accuracy evaluation agent assesses the generated answer based on its established criteria, such as relevance, accuracy, clarity, and coherence, in relation to the ground truth. A numerical score is assigned to reflect the performance of the generated response. Following this, microfluidics expert provides additional ratings to ensure the robustness and reliability of the scoring process. This combined evaluation leverages both automated and human assessments to refine the performance of the Scientific Mentor and ensure alignment with high-quality standards.

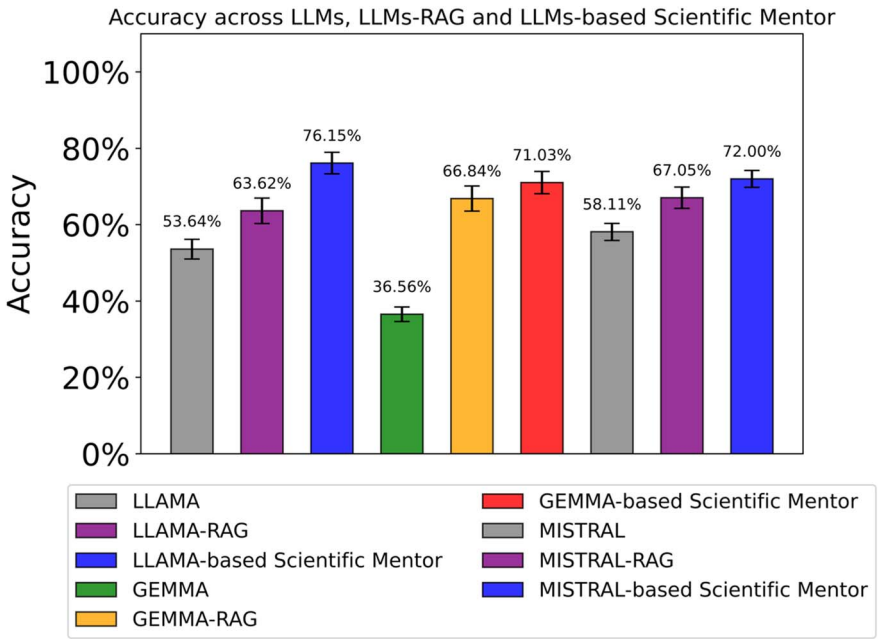
Accuracy evaluated by the accuracy evaluation agent across LLMs, LLMs-RAG and LLMs-based Scientific Mentor is compared for the QA task, as illustrated in Fig. 6a, with error bars representing the standard error. Among these, LLAMA-based Scientific Mentor demonstrate the most substantial performance gain, with accuracy rising from 53.64% in the base model to 63.62% with RAG, and further to 76.15% in its agent framework, the highest score among all configurations. This suggests that LLAMA responds particularly well to augmentation, especially within agent frameworks that allow for tool use, iterative reasoning, and task decomposition. In contrast, GEMMA in its base form achieved only 36.56% accuracy, the lowest in the set, highlighting its weaker general capabilities. Despite this, it showed the most dramatic relative gains from augmentation, reaching 66.84% with RAG and 71.03% with agent architecture. MISTRAL-based systems offered the well-balanced results across all three modes, with accuracies of 58.11% (base), 67.05% (RAG), and 72.00% (agent). Across all three LLMs, the introduction of RAG yields a significant improvement in accuracy. This reinforces the value of retrieval-

augmented methods in mitigating hallucinations and enhancing factual grounding. A critical observation is the consistently substantial accuracy gains when transitioning from standalone models to their agent-augmented counterparts, emphasizing the importance of agent-based enhancements for boosting performance. The relatively small error bars for most models suggest reliable and consistent outcomes across evaluations.

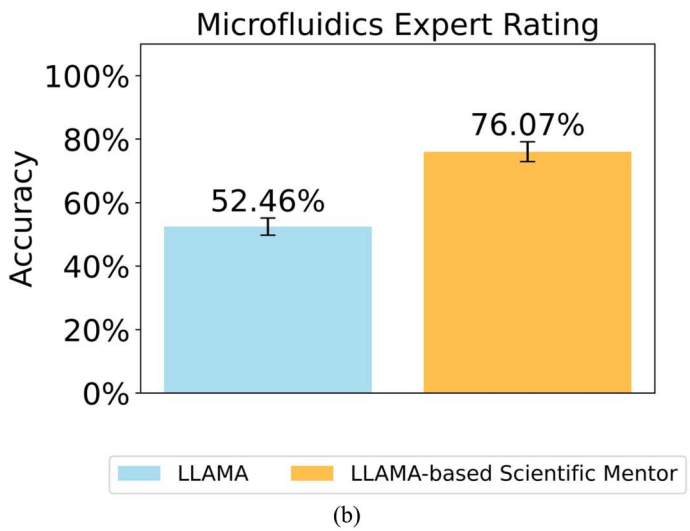
The details for performing the question–answering task with these models are provided in the SI, S3–S11.

The accuracy of LLAMA and LLAMA-based Scientific Mentor in the QA task, as rated by a microfluidics expert, is compared, with error bars indicating the standard error, as shown in Fig. 6b. LLAMA-based Scientific Mentor achieves a markedly higher accuracy of 76.07%, compared to 52.46% for LLAMA, highlighting a substantial performance gap. The inclusion of error bars indicates that this difference is statistically significant, with minimal overlap, underscoring the reliability of the results. The relatively lower accuracy of LLAMA suggests limitations in its ability to meet the requirements of expert evaluation in this domain, potentially due to deficiencies in its model architecture or training process. In contrast, the superior performance of LLAMA-based Scientific Mentor likely stems from advanced design features and optimizations that address specific shortcomings in LLAMA, making it better suited to microfluidics research.

A comprehensive performance analysis of LLAMA, GEMMA, MISTRAL, and Scientific Mentor, which is implemented using



(a)



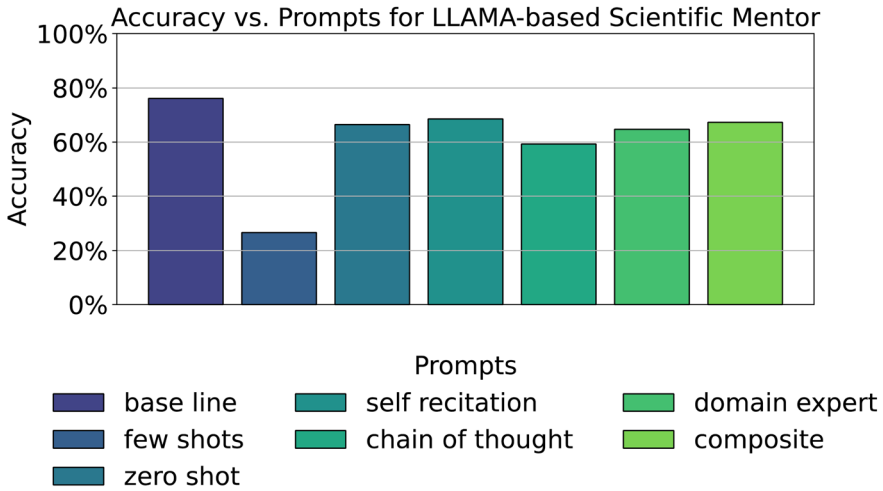
(b)

Fig. 6 (a) Accuracy evaluated by the accuracy evaluation agent across LLMs and LLMs-based Scientific Mentor for the QA task. (b) Microfluidics expert rating for QA of LLAMA-based Scientific Mentor and standalone LLAMA.

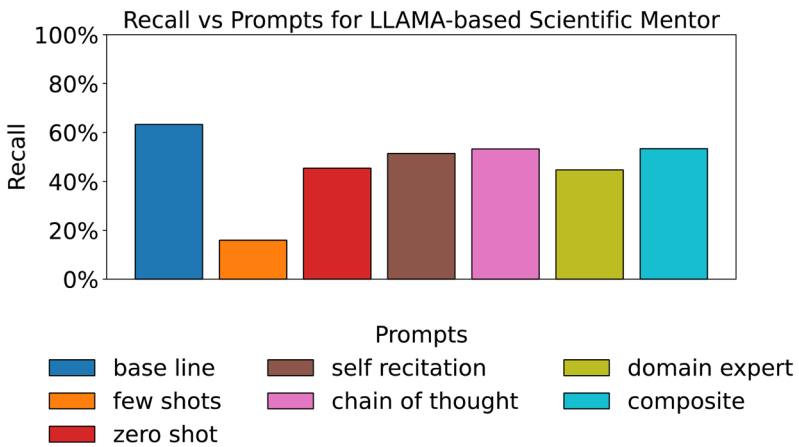
Table 6 Evaluation of extra metrics across models

	Metrics						
Model	F_1 score (%)	Recall (%)	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	METEOR
LLAMA	32.0827 ± 1.1605	30.6282 ± 1.5492	0.3268 ± 0.0117	0.0876 ± 0.0056	0.193 ± 0.0076	0.0376 ± 0.0054	0.2243 ± 0.0107
LLAMA-based Scientific Mentor	42.4434 ± 0.9869	63.2638 ± 1.4387	0.4306 ± 0.01	0.1818 ± 0.0082	0.263 ± 0.0087	0.0848 ± 0.0057	0.4328 ± 0.0117
GEMMA	16.3745 ± 1.0885	10.0868 ± 0.8011	0.1646 ± 0.0107	0.0504 ± 0.0054	0.119 ± 0.0072	0.0023 ± 0.0013	0.0861 ± 0.0056
GEMMA-based Scientific Mentor	43.6846 ± 1.08	58.5453 ± 1.2866	0.4442 ± 0.0109	0.1765 ± 0.0086	0.27 ± 0.0097	0.0806 ± 0.0063	0.4082 ± 0.0099
MISTRAL	34.0437 ± 0.866	33.8424 ± 1.0308	0.3468 ± 0.0088	0.0946 ± 0.0064	0.199 ± 0.0063	0.0445 ± 0.0052	0.2611 ± 0.0072
MISTRAL-based Scientific Mentor	39.247 ± 1.1005	51.8103 ± 1.4567	0.4003 ± 0.0112	0.1328 ± 0.0078	0.226 ± 0.0074	0.0733 ± 0.007	0.3663 ± 0.0102

LLAMA, GEMMA, and MISTRAL, is provided, across multiple performance metrics, including F_1 score, Recall, ROUGE-1, ROUGE-2, ROUGE-L, BLEU, and METEOR, each accompanied by standard errors, as shown in Table 6. The results highlight distinct trends across the models. LLAMA-based Scientific Mentor and GEMMA-based Scientific Mentor consistently



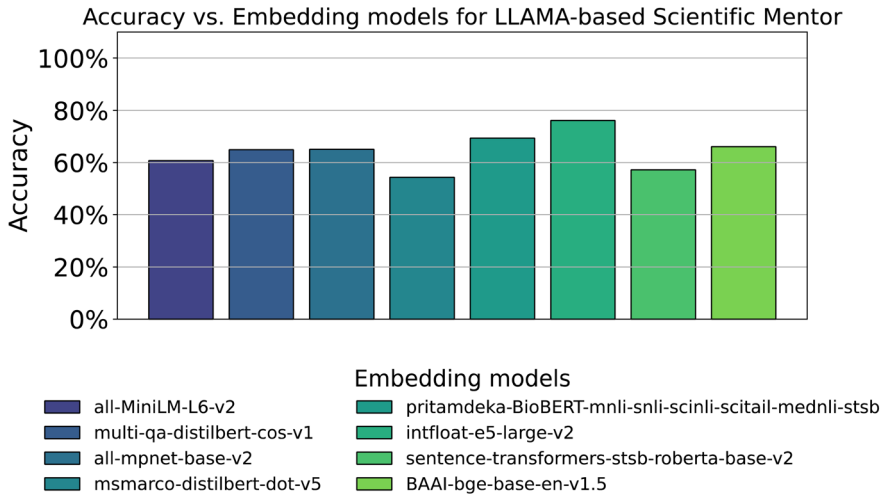
(a)



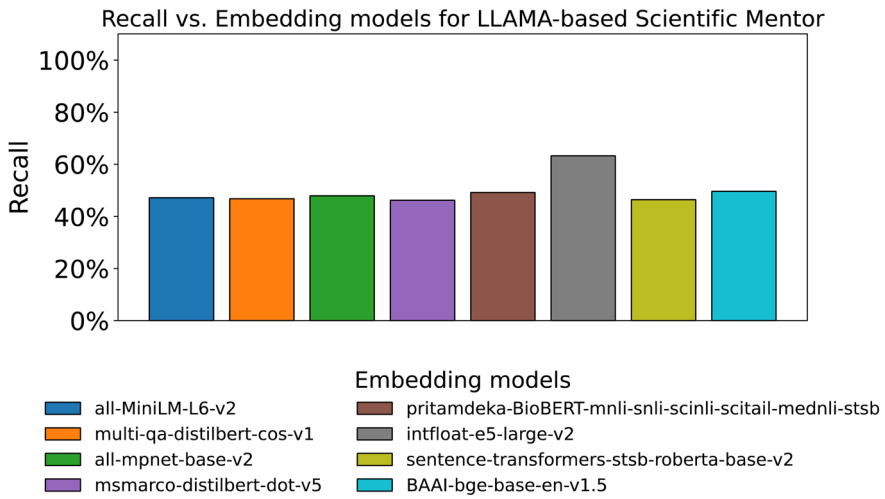
(b)

Fig. 7 (a) Accuracy *vs.* prompt for LLAMA-based Scientific Mentor. (b) Recall *vs.* prompt for LLAMA-based Scientific Mentor.

outperform their respective non-agent counterparts in nearly all metrics, showcasing the effectiveness of agent integration in these architectures. Specifically, GEMMA-based Scientific Mentor achieves the highest F_1 score (43.68 ± 1.08) and recall (58.55 ± 1.29), suggesting its strong ability to identify relevant elements in text and retrieve meaningful information. LLAMA-based Scientific Mentor also demonstrates robust performance, particularly in recall (63.26 ± 1.44), indicating its efficiency in capturing relevant data. In contrast, the significantly lower scores of GEMMA (*e.g.*, F_1 score of 16.37 ± 1.09 and ROUGE-2 of 0.0504 ± 0.0054) reflect its limitations in standalone configurations, underscoring the necessity of its agent-enhanced counterpart for improved outcomes. Similarly, MISTRAL-based Scientific Mentor outperforms MISTRAL in most metrics, with notable improvements in ROUGE-2 and METEOR, though the gap is less pronounced than the GEMMA models. Interestingly, while LLAMA-based Scientific Mentor and GEMMA-based Scientific Mentor lead in different metrics, GEMMA-based Scientific Mentor appears more balanced, maintaining high scores across all dimensions of the evaluation. The GEMMA-based Scientific Mentor's balanced performance across various metrics can be attributed to its architectural innovations and training strategies. Architecturally, GEMMA incorporates interleaved local-global attention mechanisms and grouped-query attention (GQA), enhancing both local and global context comprehension while maintaining computational efficiency.[81] These design choices enable the model to capture nuanced linguistic patterns, leading to improved performance across diverse evaluation metrics. GEMMA employs knowledge distillation from larger teacher



(a)



(b)

Fig. 8 (a) Accuracy *vs.* embedding models for LLAMA-based Scientific Mentor. (b) Recall *vs.* embedding models for LLAMA-based Scientific Mentor.

models, allowing it to learn refined representations and generalize effectively without the need for extensive parameter counts.[113–115] This approach, combined with a diverse and high-quality training dataset encompassing web documents, code, and scientific articles, equips the model with a broad knowledge base and linguistic versatility. Furthermore, the standard errors suggest that GEMMA-based Scientific Mentor and LLAMA-based Scientific Mentor yield relatively consistent results, reflecting their reliability. However, MISTRAL-based Scientific Mentor, while generally superior to MISTRAL, does not reach the performance levels of LLAMA-based Scientific Mentor or GEMMA-based Scientific Mentor.

3.2 Accuracy *vs.* prompt for LLAMA-based Scientific Mentor

The comparative performance of various prompting strategies on the accuracy of LLAMA-based Scientific Mentor as illustrated in Fig. 7a. The *y*-axis denotes classification accuracy, while the *x*-axis enumerates seven prompting paradigms: baseline, few shots, zero shot, self-recitation, chain of thought, domain expert, and composite prompts. Among these, the baseline prompt achieves the highest accuracy at approximately 76%, suggesting strong initial task performance even without sophisticated prompt engineering. Surprisingly, the few shots approach performs poorly at around 27%. Prompting strategies such as zero-shot, self-recitation, and chain-of-thought yield similar levels of accuracy ranging from 59% to 69%, indicating that reasoning-based or structured prompting offers modest benefits. Both domain expert and composite prompts slightly improve upon few shots but remain less effective than the baseline. These findings underscore the importance of tailoring prompt strategies to the Scientific Mentor's performance.

To further substantiate our findings, we assessed the impact of various prompting strategies on recall performance, as illustrated in Fig. 7b. Among the conditions tested, the baseline prompt achieved the highest recall, approximately 64%, surpassing more complex approaches such as self-recitation at around 52%, chain of thought prompting at nearly 53%, and the composite prompt at about 54%. These results suggest that the default prompting condition is already highly effective. In contrast, the few-shot prompting strategy yielded the poorest performance, with a recall of only around 16%. Intermediate recall values were observed for the zero-shot and domain expert strategies, at approximately 47 and 46%, respectively.

3.3 Accuracy *vs.* embedding models for LLAMA-based Scientific Mentor

The performance of LLAMA-based Scientific Mentor across eight embedding models as shown in Fig. 8a, with *intfloat-e5-large-v2*

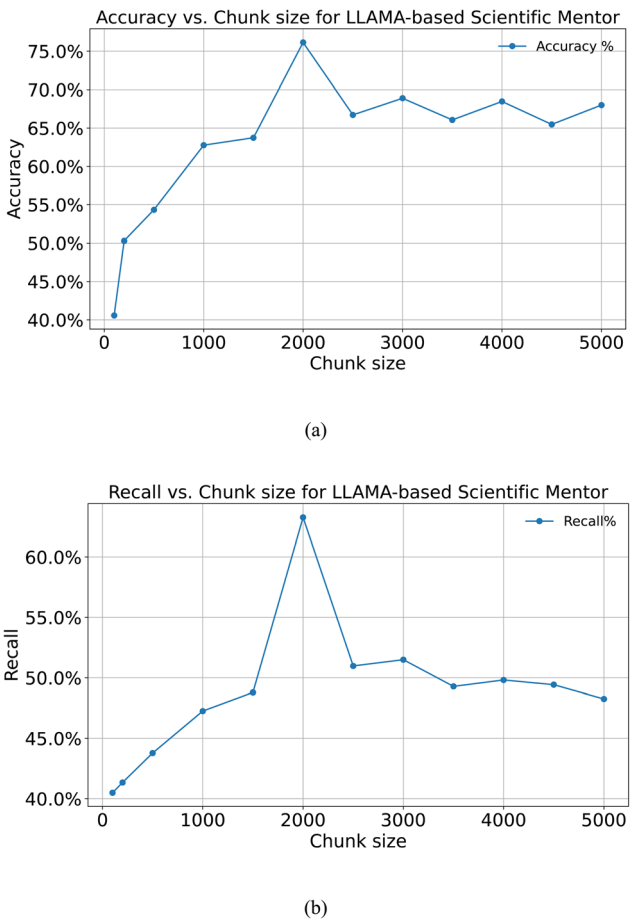


Fig. 9 (a) Accuracy *vs.* chunk size for LLAMA-based Scientific Mentor. (b) Recall *vs.* chunk size for LLAMA-based Scientific Mentor.

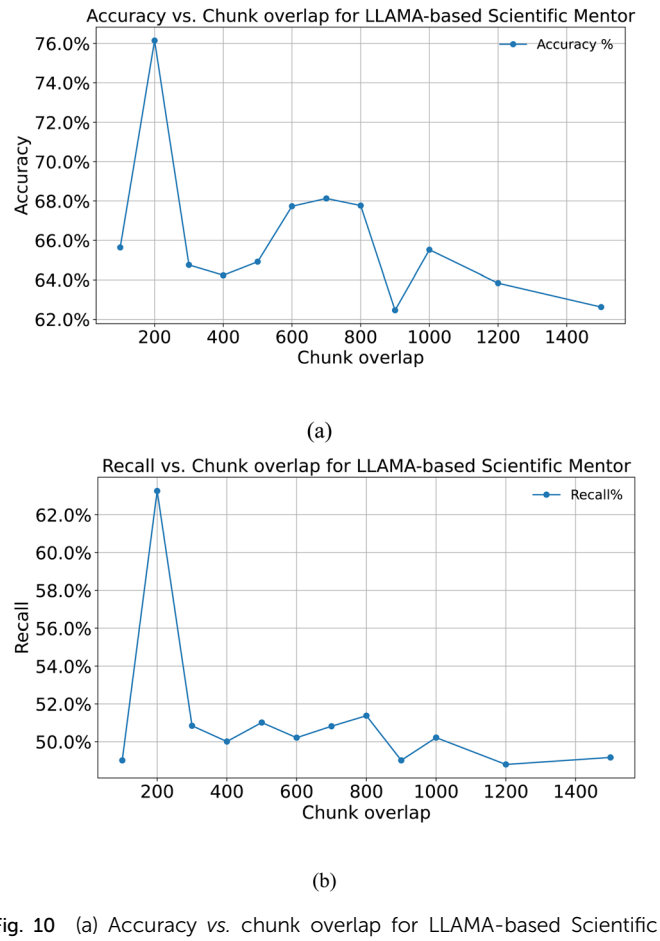


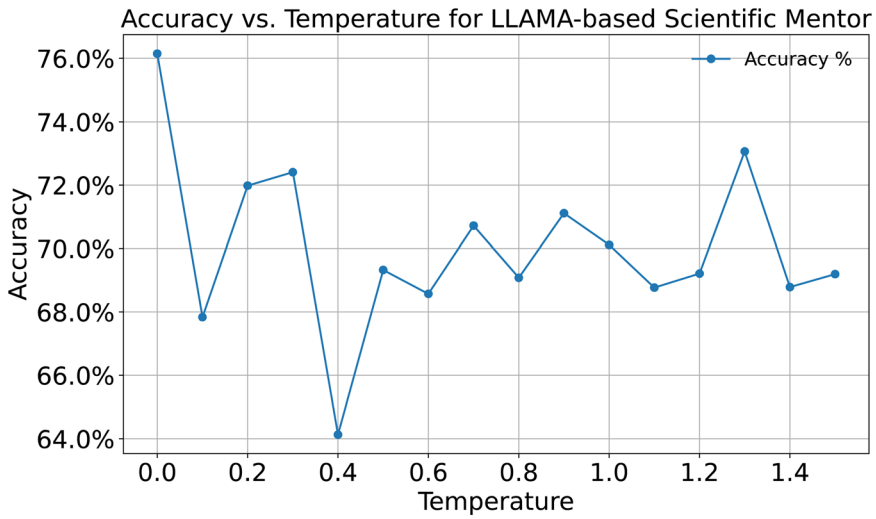
Fig. 10 (a) Accuracy *vs.* chunk overlap for LLAMA-based Scientific Mentor. (b) Recall *vs.* chunk overlap for LLAMA-based Scientific Mentor.

achieving the highest accuracy at around 76%. This result suggests that embeddings trained on large-scale ranking and retrieval tasks offer enhanced compatibility with LLAMA's reasoning mechanisms. While *pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb*, *multi-qa-distilbert-cos-v1* and *all-mpnet-base-v2* also yield commendable results, other models such as *msmarco-distilbert-dot-v5* fall short, with accuracy below 55%. The observed variation across models indicates that embedding selection has a considerable impact on the performance of the Scientific Mentor.

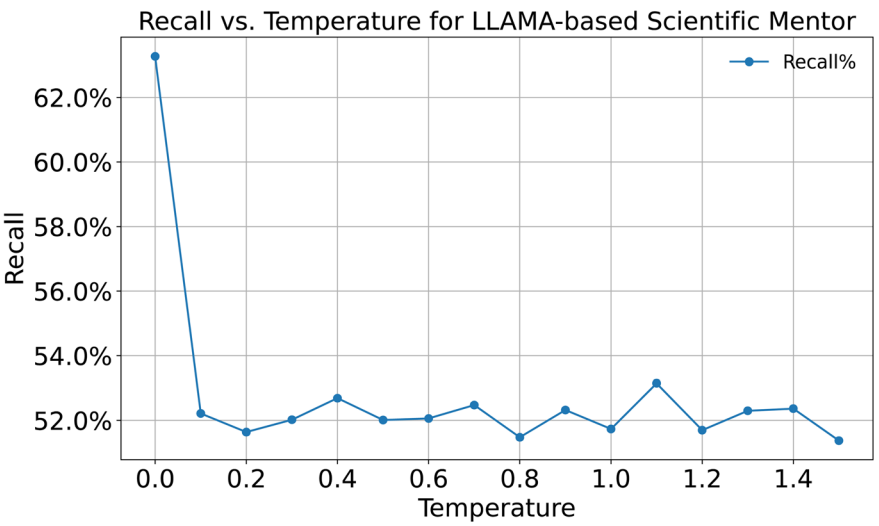
To provide additional validation for our results, we analyzed how various embedding models affect recall performance, as shown in Fig. 8b. Notably, the intfloat-e5-large-v2 model achieves the highest recall, surpassing 60%, while the remainder of the models clusters around the 45–50% range. The consistency among most models suggests a performance plateau for standard embedding strategies, whereas intfloat-e5-large-v2 demonstrates that recall can be significantly improved through advanced, larger-scale architectures.

3.4 Accuracy *vs.* chunk size for LLAMA-based Scientific Mentor

The accuracy of LLAMA-based Scientific Mentor as a function of chunk size is depicted in Fig. 9a. Accuracy surges from nearly 41% at 100 tokens to a peak of around 76% at 2000 tokens, indicating increased contextual awareness with larger inputs. However, performance exhibits a decline and oscillates across larger chunk sizes up to 5000 tokens, suggesting that beyond



(a)



(b)

Fig. 11 (a) Accuracy *vs.* temperature for LLAMA-based Scientific Mentor. (b) Recall *vs.* temperature for LLAMA-based Scientific Mentor.

a certain context length, gains in information density may be offset by increased noise. These results underscore the critical role of chunk-size optimization in maintaining the performance and computational efficiency of LLAMA-based Scientific Mentor.

To further validate our findings, we investigated the effect of varying chunk sizes on recall performance, as depicted in Fig. 9b. Initially, recall rises steadily with increasing chunk sizes, peaking dramatically near 2000, where recall reaches its highest value of approximately 64%. Beyond this threshold, the recall percentage declines. These findings indicate that excessively large chunks may reduce the ability of LLAMA-based Scientific Mentor to process and retain information effectively, emphasizing the importance of identifying an optimal chunk size for efficient operation.

3.5 Accuracy *vs.* chunk overlap for LLAMA-based Scientific Mentor

The relationship between accuracy and chunk overlap for LLAMA-based agents with a fixed chunk size of 2000 is illustrated in Fig. 10a. The accuracy peaks sharply at around 76% when the chunk overlap is 200, followed by a steep decline to approximately 64% at an overlap of 400. Subsequent increases and decreases in accuracy are observed, peaking again around 68% at overlaps of 600 and 800, before a significant drop at 900. The accuracy then fluctuates mildly and declines steadily with larger overlaps, reaching a minimum at 1500. These findings highlight the sensitivity of LLM performance to chunk overlap, emphasizing the importance of tuning overlap parameters for optimal results.

To provide additional validation for our results, we analyzed how various chunk overlaps affect recall performance, as shown in Fig. 10b. The peak recall occurs at 200-token overlap, reaching over 63%, whereas both lower and higher overlaps yield inferior performance. This trend suggests that while a modest degree of overlap facilitates effective context propagation between adjacent chunks, larger overlaps may introduce excessive redundancy, inducing contextual ambiguity. The findings imply that optimal chunk overlap should be carefully calibrated rather than maximized.

3.6 Accuracy *vs.* temperature for LLAMA-based Scientific Mentor

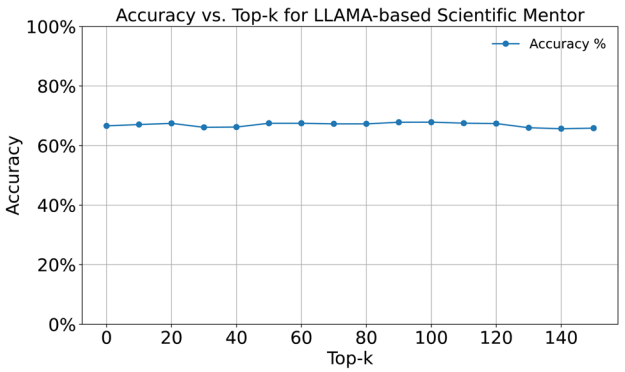
The impact of temperature on the accuracy of responses generated by LLAMA-based Scientific Mentor is illustrated in Fig. 11a. At a temperature of 0.0, a maximum accuracy of approximately 76% is recorded, but a sharp drop is observed by 0.1 and 0.4, where accuracy falls below 70%. Subsequent increases in temperature do not follow a linear or predictable path; instead, they produce oscillatory behaviour in accuracy, ranging from 64.1% to 73.3%. This instability suggests that higher temperatures do not guarantee better performance and may lead to over-randomized or incoherent outputs. The findings stress the importance of empirically identifying an optimal temperature setting to maximize the performance of LLAMA-based Scientific Mentor.

To further substantiate our results, we quantified recall performance across a range of temperature settings, as depicted in Fig. 11b. At a temperature of 0.0, the system achieves the highest recall, exceeding 63%, suggesting highly deterministic behavior enhances retrieval accuracy. However, as the temperature increases, recall sharply declines and stabilizes around 52%, indicating that increased randomness in token sampling degrades the system's ability to retrieve relevant information. The data underscore the sensitivity of LLAMA-based Scientific Mentor to sampling configurations, with elevated temperatures producing more diverse but less consistently relevant outputs.

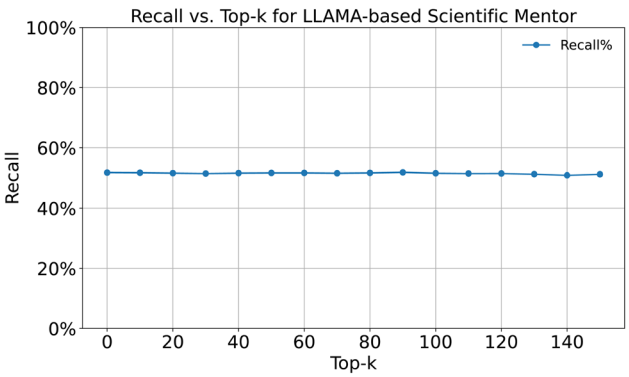
3.7 Accuracy *vs.* top-*k* for LLAMA-based Scientific Mentor

How the accuracy evolves with increasing top-*k* values in LLAMA-based Scientific Mentor is illustrated in Fig. 12a. Despite increasing the candidate set size from 0 to 150, the accuracy shows minimal variation, remaining near the 66–68% range throughout. From a deployment perspective, these findings suggest that the performance of LLAMA-based Scientific Mentor remains relatively stable across a range of top-*k* values.

To further support our findings, we examined the impact of different top-*k* configurations on recall performance, as illustrated in Fig. 12b. The curve remains notably flat, with recall values consistently near 51%, indicating that increasing the



(a)



(b)

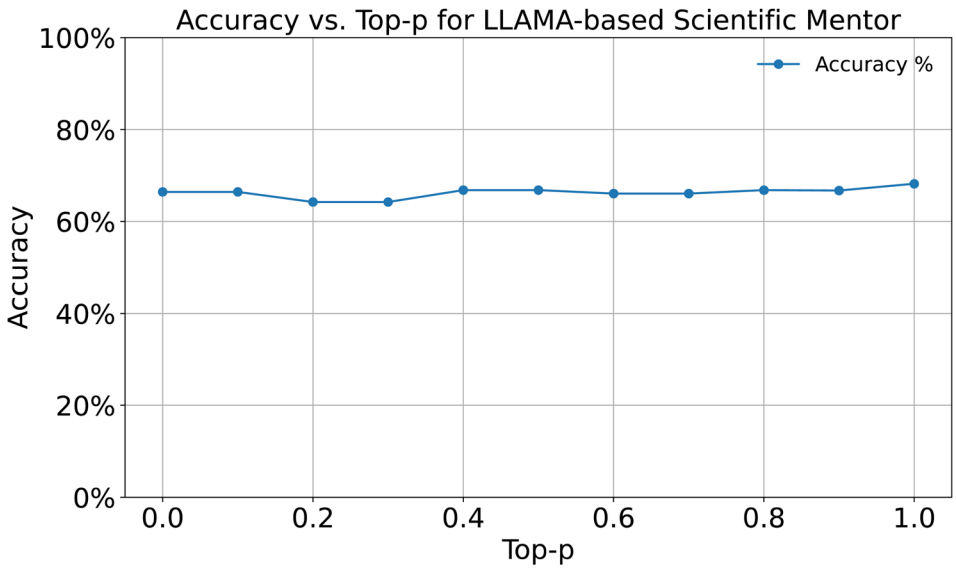
Fig. 12 (a) Accuracy *vs.* top-*k* for LLAMA-based Scientific Mentor. (b) Recall *vs.* top-*k* for LLAMA-based Scientific Mentor.

number of sampled candidates offers minimal performance gains. From an application standpoint, this suggests that LLAMA-based Scientific Mentor maintains consistent effectiveness across different top-k settings.

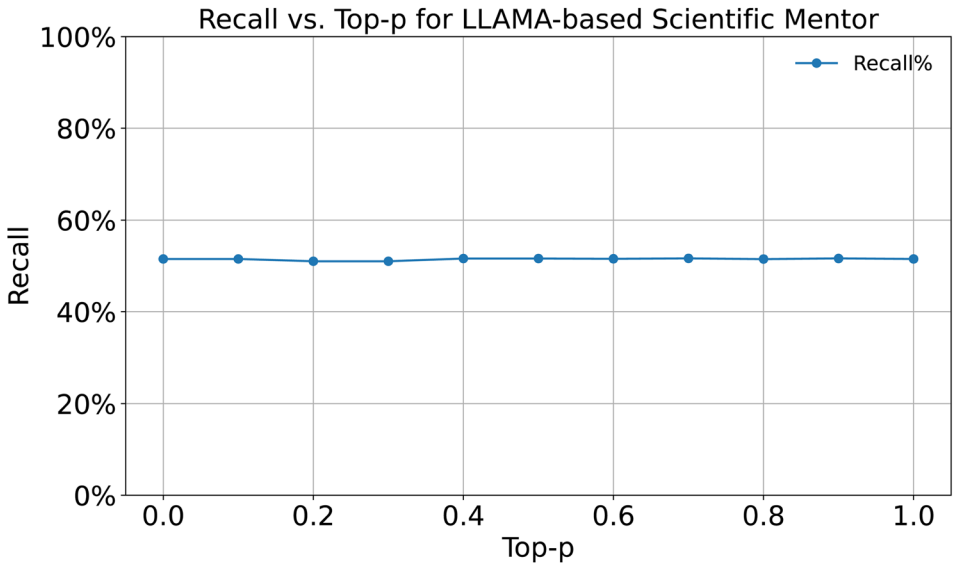
3.8 Accuracy *vs.* top-p for LLAMA-based Scientific Mentor

The relationship between accuracy and the top-p parameter for LLAMA-based Scientific Mentor is illustrated in Fig. 13a. The accuracy remains relatively stable across the top-p range from 0.0 to 1.0, with minor fluctuations around a mean value of approximately 65%. This invariance implies that the accuracy of LLAMA-based Scientific Mentor is not strongly correlated with the diversity of token selection introduced by adjusting the top-p parameter. Such findings highlight the potential robustness of LLAMA-based Scientific Mentor under diverse sampling regimes.

To reinforce our results, we assessed the effect of different top-p configurations on recall performance, as shown in Fig. 13b. The results demonstrate a remarkably stable recall rate of approximately 52% across all tested top-p values, from 0.0 to 1.0. This trend suggests that recall is not significantly impacted by the randomness introduced through nucleus sampling, highlighting the robustness of the LLAMA-based Scientific Mentor.



(a)



(b)

Fig. 13 (a) Accuracy *vs.* top-p for LLAMA-based Scientific Mentor. (b) Recall *vs.* top-p for LLAMA-based Scientific Mentor.

3.9 Performance of the Automation Designer based on LLAMA 3.1 in generating machine learning model

An example illustrates a structured workflow for generating high-quality machine learning code to predict microfluidic chip parameters, as shown in Fig. 14a. The process initiates with a query about creating the machine learning code, which is addressed by the design automation agent responsible for generating an initial version of the code. This preliminary code

Request

A .csv file with the following info.
Input feature: {Orifice width (um), Aspect ratio, Expansion ratio, Normalized orifice length, Normalized water inlet, Normalized oil inlet, Flow rate ratio, Capillary number}
Output feature: {Observed generation rate (Hz), Observed droplet diameter (um), Observed regime}
Build a XGBoost model to predict Observed generation rate (Hz). Apply hyperparameters optimization

Design Automation Agent

```
df = pd.read_csv('data.csv')
# Define features and target
X = df.drop(['Observed generation rate (Hz)'], axis=1)
y = df['Observed generation rate (Hz)']
```

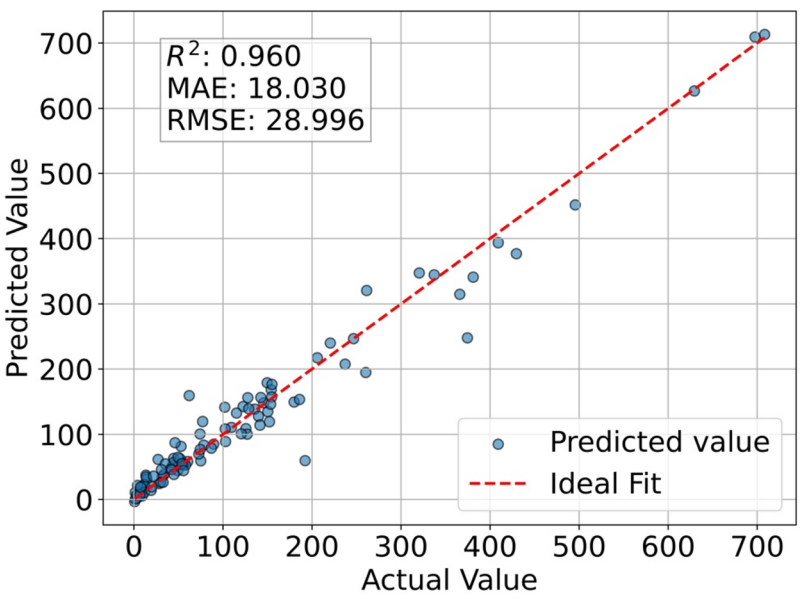
Code Reviewing Agent

* The `XGBRegressor` instance should be initialized with a dictionary of hyperparameters instead of an empty dictionary. It should look like this: `xgb_model = XGBRegressor(max_depth=3, learning_rate=0.1, n_estimators=50)`.

Code Improving Agent

```
# Initialize the XGBoost model and GridSearchCV with the specified hyperparameters
xgb_model = XGBRegressor(max_depth=3, learning_rate=0.1, n_estimators=50)
```

(a)



(b)

Fig. 14 (a) Workflow for generating machine learning code. (b) Droplet generation rate prediction.

is subsequently evaluated by the code reviewing agent, whose role is to critically analyze and identify potential issues or areas for refinement in the generated script. Feedback from this review is passed to the code improving agent, which applies the necessary modifications to enhance the functionality of the code, accuracy, and efficiency. The final stage of the process involves outputting the optimized and polished code, ready for implementation or further use (SI, S12).

An example of the predictive performance of the generated machine learning model for estimating droplet generation rates using an available flow-focusing microfluidics dataset[25] is illustrated in Fig. 14b. The scatter plot compares the predicted values with the corresponding actual values with the data points plotted against a diagonal line representing the ideal scenario where predicted values perfectly match actual values. The close clustering of most points along this diagonal suggests that the model accurately captures the underlying relationship between the input features and the droplet generation rate. Quantitative performance metrics further support this observation, with a high coefficient of determination ($R^2 = 0.96$) demonstrating

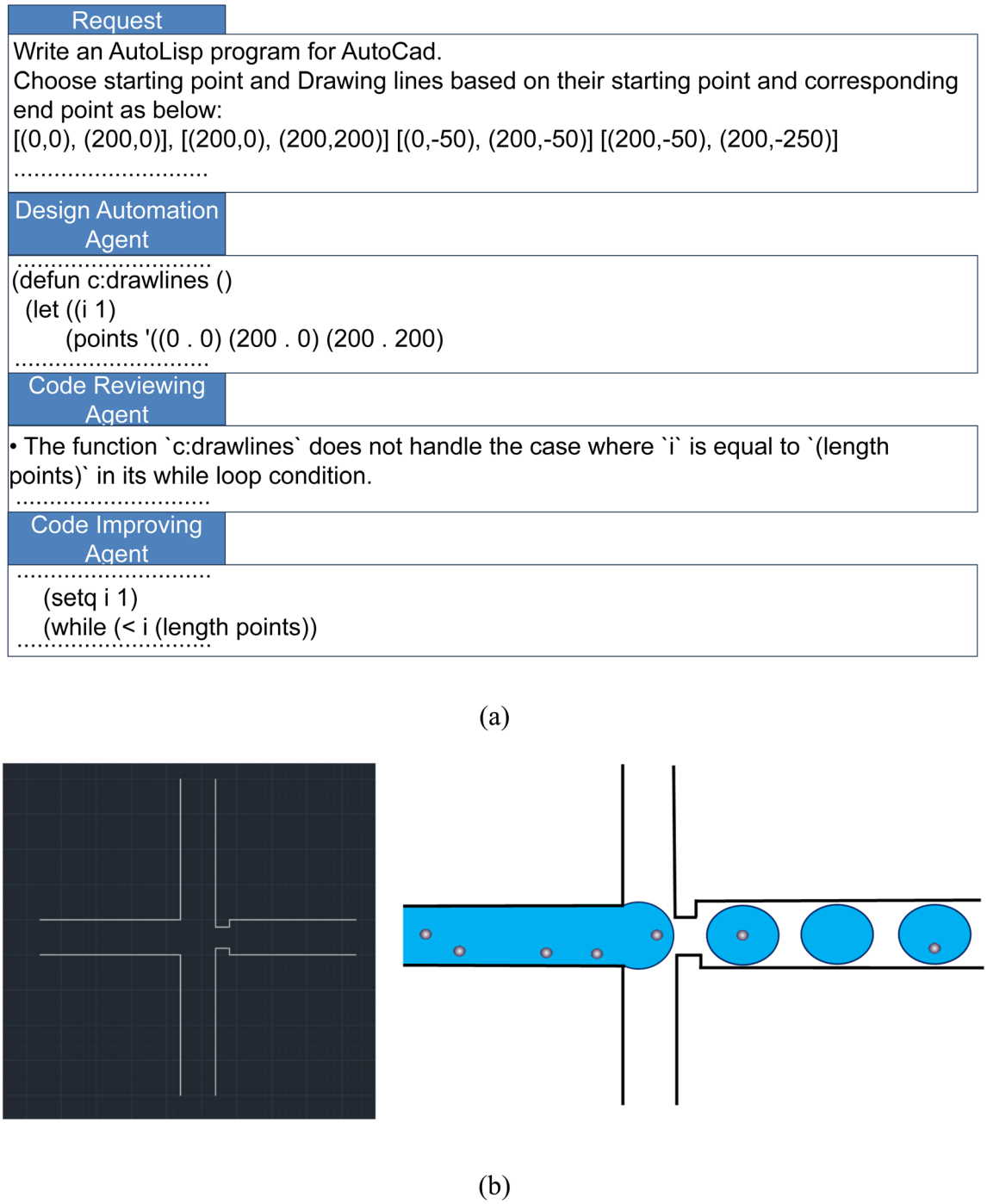


Fig. 15 (a) Workflow for generating AutoLISP code in AutoCAD. (b) Implemented AutoCAD drawing.

strong agreement between predictions and actual values. The root mean square error (RMSE) of 28.996, and the mean absolute error (MAE) of 18.03, which is lower than the MAE of 20 reported in the previous study,[25] demonstrate the effectiveness of the model in minimising prediction errors. However, some deviations are noticeable for higher actual droplet generation rates, where the predicted values slightly underestimate or overestimate the true values, suggesting potential areas for

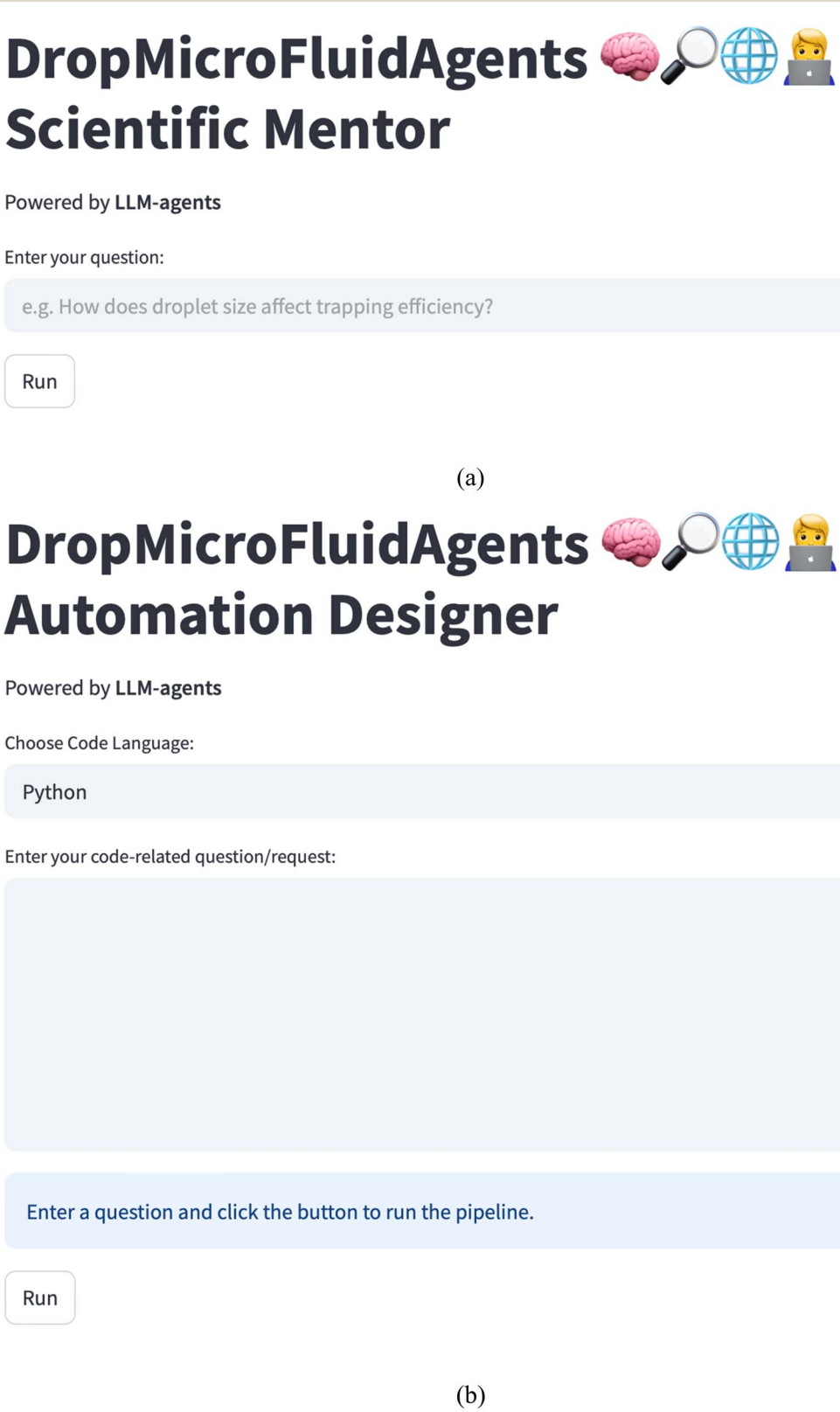


Fig. 16 GUI of DMAFs. (a) GUI of Scientific Mentor. (b) GUI of Automation Designer.

improvement in model fine-tuning. Overall, this visualisation effectively illustrates the robustness and reliability of the model in predicting droplet generation rates across a wide range of values. Hence, the Automation Designer ensures a rigorous, iterative refinement process, fostering the production of robust and reliable machine learning solutions tailored to the domain of microfluidics.

3.10 Performance of the Automation Designer based on LLAMA 3.1 in generating CAD design

An example illustrates a structured, automated pipeline for generating and refining AutoLISP code in AutoCAD to design a microfluidic chip, employing the Automation Designer, as shown in Fig. 15a. The process begins with a user posing a specific question or request regarding the AutoLISP code required for the chip design. This input is processed by the design automation agent, which generates an initial draft of the code based on the provided specifications. The generated code is then passed to the code reviewing agent, responsible for meticulously analyzing the code for errors, inconsistencies, and alignment with the design objectives. This stage ensures that the draft meets basic functional and structural requirements. Any identified issues or areas for improvement are forwarded to the code-improving agent, which refines the code further by addressing the shortcomings highlighted during the review phase. This agent optimizes the code for efficiency, accuracy, and functionality. The final, refined code is generated as the ultimate result and is ready for implementation in the design, as shown in Fig. 15b, and for the fabrication of the microfluidic chip. This automated, iterative process ensures a streamlined and reliable approach to code development, minimizing human error while maximizing precision and efficiency in microfluidic chip design (SI, S12).

3.11 An user-friendly GUI of DMAFs

To facilitate straightforward and effective user interaction, we designed a minimalistic yet intuitive GUI, as shown in Fig. 16. Comprehensive installation guidelines are available in the README documentation on the linked GitHub repository.

4. Conclusions and outlook

In this study, we present an intelligent multi-agent framework for advancing droplet microfluidics research by integrating advanced LLMs, which represents a notable milestone in the field. The framework includes the Scientific Mentor, which leverages domain-specific knowledge to provide reliable guidance on droplet microfluidics. Our experimental evaluation revealed that coupling DMFAs with the LLAMA3.1 model resulted in the highest observed accuracy of 76.15%, demonstrating the notable performance gains enabled by agent integration. The enhancement was particularly marked when DMFAs were integrated with the GEMMA2 model, yielding a 34.47% increase in accuracy compared to the standalone GEMMA2 baseline. We performed a systematic empirical investigation to assess the impact of key hyperparameters including prompting strategies, embedding models, chunk size and overlap, temperature, top-p, and top-k values on the accuracy of question–answering tasks. The baseline prompt consistently outperformed other prompting techniques. Among the embedding models evaluated, *intfloat-e5-large-v2*, noted for its sophisticated and large-scale architecture, delivered the highest accuracy. Accuracy was further enhanced by employing a simple fixed-length chunking strategy with small window sizes and minimal overlap. A temperature setting of 0.0 yielded optimal results, while adjustments to top-p and top-k values showed minimal effect. Together, these findings offer practical insights for optimizing DMFAs configurations to achieve a balance between computational efficiency and task accuracy. Additionally, the framework incorporates the Automation Designer, capable of generating machine learning code to facilitate design optimisation and automation of microfluidic chips, as well as producing code-based CAD scripts for rapid and precise creation of designs. LLMs are poised to revolutionise the field of droplet microfluidics by enhancing research methodologies and accelerating scientific discovery. Droplet microfluidics, characterised by its ability to manipulate discrete droplets in micro-scale environments, has broad applications in biotechnology, medicine, and materials science. LLM agents offer unprecedented opportunities to streamline the analysis of vast scientific literature, enabling researchers to identify patterns, generate hypotheses, and design experiments more efficiently. By providing context-aware insights and generating predictive models, these agents can aid in optimising droplet generation, stability, and functionality, significantly reducing the time and cost associated with experimental iterations. Furthermore, integrating LLMs into experimental workflows could improve reproducibility by standardising protocols, enabling researchers to detect anomalies in real time, and enhancing collaborative research by facilitating cross-disciplinary data integration. In education and training, LLM agents have the potential to transform how droplet microfluidics is taught and applied in both academic and industrial contexts. Students and professionals can leverage these agents to access interactive tutorials, troubleshoot experimental setups, and gain deeper insights into fluid dynamics and material properties. In examinations, LLMs could be utilised to design personalised assessments that evaluate conceptual understanding and problem-solving skills in droplet microfluidics. Furthermore, these agents can provide real-time feedback and adaptive learning pathways, ensuring learners at all levels can progress effectively. Additionally, these agents can bridge the gap between academia and industry by offering tailored solutions for process optimisation, quality control, and scaling up the production of microfluidic devices. By democratising access to advanced knowledge and tools, LLM agents can enable small-scale laboratories and startups to compete with larger institutions, fostering innovation across diverse settings.

Previous studies have proposed frameworks for the automation and validation of microfluidic devices, typically implemented as closed, rule-based systems.[116–118] While effective within their defined parameters, these frameworks do not

incorporate LLMs or comparable LLM agents, making it challenging to integrate digital discovery platforms at the state of the art. However, our work, droplet microfluidics LLM Agents, is capable of transforming them into closed-loop digital discovery platforms that encompass literature synthesis, hypothesis generation, autonomous design, execution in self-driving laboratories,[31,35,119–126] analysis of results, and the generation of new hypotheses.

LLM agents hold promise for advancing the automation and optimisation of computer-aided design (CAD) for microfluidic devices. The design of droplet microfluidics systems often requires precise and intricate configurations that are challenging and time-intensive to develop. By leveraging the computational power of LLMs, researchers can automate CAD processes, enabling rapid prototyping of novel device architectures. These agents can analyse complex datasets to refine channel geometries and droplet manipulation parameters, resulting in highly efficient designs. Moreover, LLMs can integrate multi-objective optimisation frameworks that balance trade-offs between cost, performance, and manufacturability, facilitating the development of robust and scalable devices. Additionally, LLMs can facilitate the implementation and optimisation of machine learning models tailored to the unique challenges of droplet microfluidics, such as predicting droplet behaviour under varying conditions, identifying emergent patterns in high-dimensional data, and optimising device performance for specific applications. This synergy between LLMs and machine learning could lead to breakthroughs in microfluidic technologies, unlocking new applications in diagnostics, drug discovery, and synthetic biology. However, integrating LLM agents into droplet microfluidics research and development is challenging. Data quality, model interpretability, and the risk of over-reliance on automated systems must be addressed. For instance, the variability in experimental conditions and datasets could lead to biased or suboptimal recommendations if not rigorously curated. Furthermore, ensuring the security of sensitive experimental data and proprietary designs is crucial in preventing intellectual property theft or misuse. Ethical concerns also arise regarding the displacement of traditional skill sets and the potential marginalisation of researchers who may lack access to LLM-driven tools. Addressing these challenges requires a concerted effort to establish best practices, including transparent validation metrics, collaborative governance frameworks, and robust training programs. Despite these hurdles, the transformative potential of LLM agents in advancing droplet microfluidics is undeniable. By bridging the gaps between computational intelligence, experimental precision, and educational accessibility, these agents promise to accelerate innovation and expand the horizons of this dynamic field.

Author contributions

Conceptualization – N.-D. D.; methodology – D.-N. N., N.-D. D.; investigation – D.-N. N.; data curation – D.-N. N.; writing – original draft – D.-N. N.; writing – review & editing – D.-N. N., N.-D. D.; supervision – R. K.-Y. T., N.-D. D.; funding acquisition – R. K.-Y. T., N.-D. D.

Conflicts of interest

There are no conflicts to declare.

Data availability

The datasets and code for the analyses and figure generations in this work are publicly available on GitHub at url: **https://github.com/duydinhlab/DMFAgents** (DOI: **https://doi.org/10.5281/zenodo.16875111**).

The supplementary information includes the framework outputs, the ground-truth QA dataset, and the collection of PDF documents used for database construction. See DOI: **https://doi.org/10.1039/d5dd00306g**.

Acknowledgements

We gratefully acknowledge the funding provided by the Research Grant Council of Hong Kong, General Research Fund (Ref No. 14211223).

References

1 G. M. Whitesides, *Nature*, 2006, **442**, 368–373.
2 Y. Ding, P. D. Howes and A. J. Demello, *Anal. Chem.*, 2020, **92**, 132–149.
3 E. Y. u. Basova and F. Foret, *Analyst*, 2014, **140**, 22–38.
4 T. Moragues, *et al.*, *Nat. Rev. Methods Primers*, 2023, **3**, 1–22.
5 N. D. Dinh, *et al.*, *Lab Chip*, 2013, **13**, 1402–1412.
6 N. D. Dinh, *et al.*, *Small*, 2017, **13**, 1700684.
7 H. Tan, S. Guo, N. D. Dinh, R. Luo, L. Jin and C. H. Chen, *Nat. Commun.*, 2017, **8**, 1–10.
8 N. D. Dinh, *et al.*, *Lab Chip*, 2020, **20**, 2756–2764.
9 N.-D. Dinh, *et al.*, *arXiv*, 2024, preprint, arXiv:2501.01962v1, DOI: **10.48550/arXiv.2501.01962v1**.
10 R. Zilionis, *et al.*, *Nat. Protoc.*, 2016, **12**, 44–73.
11 M. Pellegrino, *et al.*, *Genome Res.*, 2018, **28**, 1345–1352.
12 X. Zhang, *et al.*, *Mol. Cell*, 2019, **73**(1), 130–142.e5.
13 A. M. Klein, *et al.*, *Cell*, 2015, **161**, 1187–1201.
14 E. Z. Macosko, *et al.*, *Cell*, 2015, **161**, 1202–1214.
15 A. Gérard, *et al.*, *Nat. Biotechnol.*, 2020, **38**, 715–721.
16 K. Fischer, *et al.*, *Nat. Biotechnol.*, 2025, **43**, 960–970.
17 J. J. Agresti, *et al.*, *Proc. Natl. Acad. Sci. U. S. A.*, 2010, **107**, 4004–4009.
18 H. Yin, *et al.*, *Biosens. Bioelectron.*, 2021, **188**, 113282.
19 Y. Belotti and C. T. Lim, *Anal. Chem.*, 2021, **93**, 4727–4738.
20 T. S. Kaminski, O. Scheler and P. Garstecki, *Lab Chip*, 2016, **16**, 2168–2187.
21 D. T. Chiu, *et al.*, *Chem*, 2017, **2**, 201–223.
22 F. Su, K. Chakrabarty and R. B. Fair, *IEEE Trans. Comput. Aided Des. Integrated Circ. Syst.*, 2006, **25**, 211–223.
23 S. Battat, D. A. Weitz and G. M. Whitesides, *Lab Chip*, 2022, **22**, 530–536.

24 D. McIntyre, A. Lashkaripour, P. Fordyce and D. Densmore, *Lab Chip*, 2022, **22**, 2925–2937.
25 A. Lashkaripour, *et al.*, *Nat. Commun.*, 2021, **12**, 1–14.
26 D. Nguyen, R. K. Tong and N. Dinh, *arXiv*, 2024, preprint, arXiv:2411.06691v1, DOI: 10.48550/arXiv.2411.06691v1.
27 S. A. Damiati, *et al.*, *Sci. Rep.*, 2020, **10**, 1–11.
28 S. H. Hong, H. Yang and Y. Wang, *Microfluid. Nanofluid.*, 2020, **24**, 1–20.
29 W. Ji, T. Y. Ho, J. Wang and H. Yao, *IEEE Trans. Comput. Aided Des. Integrated Circ. Syst.*, 2020, **39**, 2544–2557.
30 T. Savage, *et al.*, *Nat. Chem. Eng.*, 2024, **1**, 522–531.
31 A. A. Volk, *et al.*, *Nat. Commun.*, 2023, **14**, 1–16.
32 Y. Pan, *et al.*, *Chem. Eng. Sci.*, 2025, **311**, 121567.
33 J. Wei, *et al.*, *arXiv*, 2022, preprint, arXiv:2206.07682v2, DOI: 10.48550/arXiv.2206.07682v2.
34 Z. Zheng, *et al.*, *Angew Chem. Int. Ed. Engl.*, 2023, **62**(46), 1–8.
35 D. A. Boiko, R. MacKnight, B. Kline and G. Gomes, *Nature*, 2023, **624**, 570–578.
36 Z. Zheng, O. Zhang, C. Borgs, J. T. Chayes and O. M. Yaghi, *J. Am. Chem. Soc.*, 2023, **145**, 18048–18062.
37 K. M. Jablonka, P. Schwaller, A. Ortega-Guerrero and B. Smit, *Nat. Mach. Intell.*, 2024, **6**, 161–169.
38 A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White and P. Schwaller, *Nat. Mach. Intell.*, 2024, **6**, 525–535.
39 W. Hou and Z. Ji, *Nat. Methods*, 2024, **21**, 1462–1465.
40 H. Cui, C. Wang, H. Maan, K. Pang, F. Luo, N. Duan and B. Wang, *Nat. Methods*, 2024, **21**, 1470–1480.
41 R. Riveland and A. Pouget, *Nat. Neurosci.*, 2024, **27**, 988–999.
42 Z. Lin, *et al.*, *Science*, 2023, **379**, 1123–1130.
43 Z. Xiao, W. Li, H. Moon, G. W. Roell, Y. Chen and Y. J. Tang, *ACS Synth. Biol.*, 2023, **12**, 2973–2982.
44 J. Lee, *et al.*, *Bioinformatics*, 2019, **36**, 1234–1240.
45 Q. Chen, *et al.*, *Bioinformatics*, 2023, **39**(9), 1–9.
46 S. Thapa and S. Adhikari, *Ann. Biomed. Eng.*, 2023, **51**, 2647–2651.
47 Q. Jin, Y. Yang, Q. Chen and Z. Lu, *Bioinformatics*, 2023, **40**(2), 1–8.
48 X. Zhang, Z. Zhou, C. Ming and Y. Y. Sun, *J. Phys. Chem. Lett.*, 2023, **14**, 11342–11349.
49 Z. Hong, *Energy Mater. Adv.*, 2023, **4**, 1–3.
50 M. P. Polak and D. Morgan, *Nat. Commun.*, 2024, **15**, 1–11.
51 J. Choi and B. Lee, *Commun. Mater.*, 2024, **5**, 1–11.
52 O. N. Oliveira, L. Christino, M. Oliveira and F. V. Paulovich, *J. Chem. Inf. Model.*, 2023, **63**, 7605–7609.
53 T. Xie, *et al.*, *arXiv*, 2023, preprint, arXiv:2304.02213v5, DOI: 10.48550/arXiv.2311.07361v2.
54 M. Zaki, *et al.*, *Digital Discovery*, 2024, **3**, 313–327.
55 D. Van Veen, *et al.*, *Nat. Med.*, 2024, **30**, 1134–1142.
56 F. Wong, C. de la Fuente-Nunez and J. J. Collins, *Science*, 2023, **381**, 164–170.
57 M. Moor, *et al.*, *Nature*, 2023, **616**, 259–265.
58 R. Wang, H. Feng and G. W. Wei, *J. Chem. Inf. Model.*, 2023, **63**, 7189–7209.
59 A. J. Thirunavukarasu, *et al.*, *Nat. Med.*, 2023, **29**, 1930–1940.
60 Y. Wang, Y. Zhao and L. Petzold, *Proc. Mach. Learn. Res.*, 2023, vol. 219, pp. 804–823.
61 H. Wang, *et al.*, *Nature*, 2023, **620**, 47–60.
62 C. Stokel-Walker and R. Van Noorden, *Nature*, 2023, **614**, 214–216.
63 K. M. Merz, G. W. Wei and F. Zhu, *J. Chem. Inf. Model.*, 2023, **63**, 5395.
64 M. R. AI4Science and M. A. Quantum, *arXiv*, 2023, preprint, arXiv:2311.07361v2, DOI: 10.48550/arXiv.2311.07361v2.
65 Y. Zhang, *et al.*, *arXiv*, 2023, preprint, arXiv:2309.01219v2, DOI: 10.48550/arXiv.2309.01219v2.
66 M. Sallam, *Healthcare*, 2023, **11**(6), 1–20.
67 P. Lewis, *et al.*, *Adv. Neural Inf. Process. Syst.*, 2020, 9459–9474.
68 Y. Gao, *et al.*, *arXiv*, 2023, preprint, arXiv:2312.10997v5, DOI: 10.48550/arXiv.2312.10997v5.
69 J. Li, *et al.*, *arXiv*, 2024, preprint, arXiv:2402.05120v2, DOI: 10.48550/arXiv.2402.05120v2.
70 J. S. Park, *et al.*, *UIST 2023 – Proc. 36th Annu. ACM Symp. User Interface Softw. Technol.*, 2023.
71 L. Wang, *et al.*, *Front. Comput. Sci.*, 2023, **18**(6), 1–26.
72 A. Zhao, *et al.*, *Proc. AAAI Conf. Artif. Intell.*, 2023, **38**, 19632–19642.
73 T. Guo, *et al.*, *arXiv*, 2024, preprint, arXiv:2402.01680v2, DOI: 10.48550/arXiv.2402.01680v2.
74 C. Gao, *et al.*, *Humanit. Soc. Sci. Commun.*, 2024, **11**, 1–24.
75 M. D. Skarlinski, *et al.*, *arXiv*, 2024, preprint, arXiv:2409.13740v2, DOI: 10.48550/arXiv.2409.13740v2.
76 J. Li, *et al.*, *arXiv*, 2024, preprint, arXiv:2405.02957v1, DOI: 10.48550/arXiv.2405.02957v1.
77 X. Tang, *et al.*, *arXiv*, 2024, preprint, arXiv:2311.10537v4, DOI: 10.48550/arXiv.2311.10537v4.
78 S. Schmidgall, *et al.*, *arXiv*, 2024, preprint, arXiv:2405.07960v4, DOI: 10.48550/arXiv.2405.07960v4.
79 A. Dubey, *et al.*, *arXiv*, 2024, preprint, arXiv:2407.21783v2, DOI: 10.48550/arXiv.2407.21783v2.
80 A. Q. Jiang, *et al.*, *arXiv*, 2023, preprint, arXiv: 2310.06825v1, DOI: 10.48550/arXiv.2310.06825v1.
81 G. Team, *et al.*, *arXiv*, 2024, preprint, arXiv:2408.00118v3, DOI: 10.48550/arXiv.2408.00118v3.
82 Y. Li, *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, 2023, pp. 641–647.
83 L. Wang, *et al.*, *npj Digit. Med.*, 2024, **7**, 1–9.
84 P. Liu, *et al.*, *ACM Comput. Surv.*, 2021, **55**(9), 1–35.
85 L. Reynolds and K. McDonell, *arXiv*, 2021, preprint, arXiv:2102.07350v1, DOI: 10.48550/arXiv.2102.07350v1.
86 T. B. Brown, *et al.*, *arXiv*, 2020, preprint, arXiv:2005.14165v4, DOI: 10.48550/arXiv.2005.14165v4.
87 J. White, *et al.*, *arXiv*, 2023, preprint, arXiv:2302.11382v1, DOI: 10.48550/arXiv.2302.11382v1.
88 Z. Sun, *et al.*, *arXiv*, 2022, preprint, arXiv:2210.01296v2, DOI: 10.48550/arXiv.2210.01296v2.
89 T. Kojima, *et al.*, *arXiv*, 2022, preprint, arXiv:2205.11916v4, DOI: 10.48550/arXiv.2205.11916v4.
90 J. Wei, *et al.*, *arXiv*, 2022, preprint, arXiv:2201.11903v6, DOI: 10.48550/arXiv.2201.11903v6.
91 S. Huo, *et al.*, *arXiv*, 2023, preprint, arXiv:2309.11392v1, DOI: 10.48550/arXiv.2309.11392v1.

92 R. Wang, *et al.*, *arXiv*, 2023, preprint, arXiv:2305.13733v2, DOI: 10.48550/arXiv.2305.13733v2.
93 H. Su, *et al.*, *arXiv*, 2022, preprint, arXiv:2212.09741v3, DOI: 10.48550/arXiv.2212.09741v3.
94 L. Wang, *et al.*, *arXiv*, 2024, preprint, arXiv:2402.05672v1, DOI: 10.48550/arXiv.2402.05672v1.
95 L. Caspari, *et al.*, *arXiv*, 2024, preprint, arXiv:2407.08275v1, DOI: 10.48550/arXiv.2407.08275v1.
96 L. Wang, *et al.*, *arXiv*, 2022, preprint, arXiv:2212.03533v2, DOI: 10.48550/arXiv.2212.03533v2.
97 J. Lee, *et al.*, *arXiv*, 2019, preprint, arXiv:1901.08746v4, DOI: 10.48550/arXiv.1901.08746v4.
98 Y. Liu, *et al.*, *arXiv*, 2019, preprint, arXiv:1907.11692v1, DOI: 10.48550/arXiv.1907.11692v1.
99 S. R. Bhat, *et al.*, *arXiv*, 2025, preprint, arXiv:2505.21700v2, DOI: 10.48550/arXiv.2505.21700v2.
100 G. Xiong, *et al.*, *arXiv*, 2024, preprint, arXiv:2402.13178v2, DOI: 10.48550/arXiv.2402.13178v2.
101 K. Juvekar and A. Purwar, *arXiv*, 2024, preprint, arXiv:2407.19794v2, DOI: 10.48550/arXiv.2407.19794v2.
102 A. J. Yepes, *et al.*, *arXiv*, 2024, preprint, arXiv:2402.05131v3, DOI: 10.48550/arXiv.2402.05131v3.
103 A. Ammar, *et al.*, *arXiv*, 2025, preprint, arXiv:2505.08445v1, DOI: 10.48550/arXiv.2505.08445v1.
104 M. Renze and E. Guven, *arXiv*, 2024, preprint, arXiv:2402.05201v3, DOI: 10.48550/arXiv.2402.05201v3.
105 C. Wang, S. X. Liu and A. H. Awadallah, *arXiv*, 2023, preprint, arXiv:2303.04673v2, DOI: 10.48550/arXiv.2303.04673v2.
106 P.-H. Wang, *et al.*, *arXiv*, 2020, preprint, arXiv:2012.13575v1, DOI: 10.48550/arXiv.2012.13575v1.
107 G. Hinton, *et al.*, *arXiv*, 2015, preprint, arXiv:1503.02531v1, DOI: 10.48550/arXiv.1503.02531v1.
108 M. Peeperkorn, *et al.*, *arXiv*, 2024, preprint, arXiv:2405.00492v1, DOI: 10.48550/arXiv.2405.00492v1.
109 D. Patel, *et al.*, *medRxiv*, 2024, preprint, DOI: 10.1101/2024.07.22.24310824.
110 A. Holtzman, *et al.*, *arXiv*, 2019, preprint, arXiv:1904.09751v2, DOI: 10.48550/arXiv.1904.09751v2.
111 M. N. Nguyen, *et al.*, *arXiv*, 2024, preprint, arXiv:2407.01082v6, DOI: 10.48550/arXiv.2407.01082v6.
112 C. Shi, *et al.*, *arXiv*, 2024, preprint, arXiv:2402.06925v3, DOI: 10.48550/arXiv.2402.06925v3.
113 H. Bansal, *et al.*, *arXiv*, 2024, preprint, arXiv:2408.16737v2, DOI: 10.48550/arXiv.2408.16737v2.
114 D. Nadeau, *et al.*, *arXiv*, 2024, preprint, arXiv:2404.09785v1, DOI: 10.48550/arXiv.2404.09785v1.
115 M. P. Priola, *et al.*, *arXiv*, 2024, preprint, arXiv:2412.04235v2, DOI: 10.48550/arXiv.2412.04235v2.
116 P. Ebner and R. Wille, *Proc. IEEE Comput. Soc. Annu. Symp. VLSI, ISVLSI*, 2024, pp. 278–283.
117 M. Emmerich, P. Ebner and R. Wille, *IEEE Trans. Comput. Aided Des. Integrated Circ. Syst.*, 2025, **44**, 2287–2299.
118 R. Wille, B. Li, R. Drechsler and U. Schlichtmann, *Forum Specif. Des. Lang.*, Garching, Germany, 2018, pp. 5–16.
119 H. Tao, T. Wu, S. Kheiri, M. Aldeghi, A. Aspuru-Guzik, E. Kumacheva, H. Tao, T. Wu, M. Aldeghi, A. Aspuru-Guzik, E. Kumacheva and S. Kheiri, *Adv. Funct. Mater.*, 2021, **31**, 2106725.
120 R. W. Epps, A. A. Volk, K. G. Reyes and M. Abolhasani, *Chem. Sci.*, 2021, **12**, 6025–6036.
121 M. Abolhasani and E. Kumacheva, *Nature Synthesis*, 2023, **2**, 483–492.
122 S. Sadeghi, K. Mattsson, J. Glasheen, V. Lee, C. Stark, P. Jha, N. Mukhin, J. Li, A. Ghorai, N. Orouji, C. H. J. Moran, A. Velayati, J. A. Bennett, R. B. Canty, K. G. Reyes and M. Abolhasani, *Digital Discovery*, 2025, **4**, 1722–1733.
123 H. Hysmith, E. Foadian, S. P. Padhy, S. V. Kalinin, R. G. Moore, O. S. Ovchinnikova and M. Ahmadi, *Digital Discovery*, 2024, **3**, 621–636.
124 R. B. Canty, J. A. Bennett, K. A. Brown, T. Buonassisi, S. V. Kalinin, J. R. Kitchin, B. Maruyama, R. G. Moore, J. Schrier, M. Seifrid, S. Sun, T. Vegge and M. Abolhasani, *Nat. Commun.*, 2025, **16**, 1–11.
125 A. Ghafarollahi and M. J. Buehler, *Digital Discovery*, 2024, **3**, 1389–1409.
126 T. K. Chan and N.-D. Dinh, *medRxiv*, 2025, preprint, DOI: 10.1101/2025.01.01.25319863.