Microfluidic design optimization in the era of AI agents

Ivana Kundačina [1*], Ognjen Kundačina [2], Nikolina Janković [1], Dragiša Mišković [3], Vasa Radonić [1]

[1] University of Novi Sad, BioSense Institute, Dr Zorana Đinđića 1, 21000 Novi Sad, Serbia
[2] Microsoft Corporation, Bulevar Mihajla Pupina 6, 11000 Belgrade, Serbia
[3] The Institute for Artificial Intelligence Research and Development of Serbia, Fruškogorska 1, 21000 Novi Sad, Serbia
*Corresponding author: Ivana Kundačina, ivana.kundacina@biosense.rs

Abstract

Artificial intelligence (AI) is increasingly used to optimize design of microfluidic devices by predicting their performance, reducing the number of simulations, and guiding the search for design configurations with improved performances. However, in most studies, the design problem is still defined by the researcher. The device concept, design variables, parameter ranges, objective functions, and constraints are selected in advance, while AI is used to search within these predefined limits. A key limitation of current approaches is that AI is involved only in a specific step of the workflow, performing a narrowly defined task. The emergence of AI agents may change this workflow more fundamentally. AI agents are autonomous systems capable of coordinating computational processes and adapting their behaviour to achieve a defined goal. In other words, in an agentic design workflow, an AI system could interpret an optimization goal, select computational tools, define or modify the design space, run and analyse numerical simulations, and adjust the optimization strategy based on the physical behaviour of the system. This perspective examines how AI agents could extend microfluidic design optimization beyond fixed, researcher-defined search spaces toward adaptive and physics-grounded workflows. We envision that future microfluidic design optimization will use a range of methods with different levels of AI-agent involvement, from limited assistance to more adaptive and autonomous workflows. The distinctive contribution of this perspective is to provide a unified framework for comparing these approaches according to the level and role of AI-agent involvement in the design process. Accordingly, we discuss a spectrum of agentic approaches, ranging from LLM-driven conceptual design to agent-supervised numerical optimization, together with evolving role of human supervision and technical challenges associated with reliable implementation. Moving from fixed optimization workflows towards agent-guided design could enable microfluidic systems to be explored more efficiently and may shift design optimization from simple parameter search towards a more adaptive and reasoning-based process.

Keywords: AI agents, large language models, artificial intelligence, machine learning, microfluidics, CFD, design optimization

1 Introduction

Microfluidic systems have emerged as powerful platforms for the precise manipulation of small fluid volumes, offering reduced sample and reagent consumption, rapid analysis, high sensitivity, and the possibility of integrating multiple laboratory operations within compact and portable devices [1,2]. These advantages have enabled applications in chemical analysis [3,4], biomedical diagnostics [5,6], organ-on-a-chip [7–9], and lab-on-a-chip technologies [10,11]. Despite their broad potential, the design of microfluidic devices remains a complex task [12], since device performance is often strongly affected by channel geometry, operating conditions, fluid properties, and their interactions. Even small changes in these parameters can alter flow behaviour and lead to a significant change in the final device performance. As a result, the development of an efficient microfluidic system commonly requires repeated cycles of design, simulation, fabrication, and experimental testing.

Numerical simulations have therefore become an important part of microfluidic design process, allowing device performance to be evaluated before fabrication and thereby saving time and resources [13–15]. In that sense, computational fluid dynamics (CFD) is widely used to study velocity and pressure fields, as well as processes such as mixing, particle separation, droplet generation, or other transport phenomena. However, as the number of geometric and operating parameters increases, the number of possible design combinations expands exponentially. Extensive simulation strategies and manual parameter variation can become computationally demanding while still potentially leaving relevant regions of the design space unexplored. This has encouraged the use of optimization methods to guide the search towards promising designs more efficiently.

Machine learning (ML) has further expanded the computational tools available for microfluidic design and operation [16,17]. In droplet microfluidics [18,19], deep neural networks (DNNs) have been used to model droplet-flow behavior [20], convolutional autoencoders to predict droplet stability [21], residual block and Fourier enhanced networks to predict droplet size and generation frequency [22], and convolutional neural networks to characterize mixing in droplets formed by binary coalescence [23]. DNN-based approaches have also been applied to bubble detection in microfluidic channels used for point-of-care diagnostics [24], as well as to bubble detection and tracking in gas–liquid exchange processes [25]. In microfluidic biosensing [26], ML has been applied to predict device performance and optimize geometrical and operating parameters, including the response time of rotating microfluidic biosensors [27], the geometry of microchannels and sensing regions [28], and flow- and transport-related parameters affecting biosensor response [29].

ML has also been introduced into microfluidic manufacturing and transport-related applications. ML-assisted manufacturing optimization is particularly challenging because fabrication outcomes depend on multiple coupled process parameters, material properties, and machine-specific effects. Recent studies have demonstrated that ML can help address this complexity. Wang et al. [30] used a Random Forest model to optimize DLP printing parameters and reduce dimensional errors in printed microfluidic structures, while Norouzi et al. [31] applied ML to link CNC micromilling parameters, surface roughness, and hydraulic performance in PMMA microchannels. These examples show that ML can support the selection of manufacturing parameters before fabrication, although its reliability remains dependent on representative experimental data and the specific manufacturing platform. Moreover, decision-tree models have been employed to identify suitable operating conditions for nanoparticle synthesis in microfluidic reactors [32], while tree-based ensemble methods have been used to predict viscous-fingering phenomena in Hele-Shaw cells and characterize the onset of flow instabilities [33].

In such cases, ML accelerates or replaces simulations or experiments but does not change how the design problem itself is formulated. Design automation goes further by allowing computational methods to generate, configure, or optimize device designs toward a specified target. Beyond performance analysis, ML has also been applied to automate selected operational and design-related tasks. For example, reinforcement learning has been applied to the automated flow control of both continuous-flow and segmented-flow microfluidic systems and digital microfluidic biochips synthesis in selecting reliable routing configurations [34,35]. Besides reinforcement learning, other approaches use Bayesian decision-making incorporated into digital microfluidic biochip synthesis to account for uncertainty in the design space [36]. Birtek et al. [37] developed an ML-based framework for automated microfluidic circuit design, while Khalghollah et al. [38] combined CFD, a neural-network surrogate, and numerical optimization to automatically configure capillary microfluidic geometries for target concentration profiles.

These studies demonstrate that ML can support microfluidic systems at several levels, including image and signal interpretation, performance prediction, process control, uncertainty management, and design optimization. Nevertheless, in many existing applications, ML is used to address a narrowly defined task within a workflow whose structure has already been established by the researcher. The device concept, input variables, operating ranges, model outputs, and evaluation criteria generally remain fixed, while ML is used to approximate, classify, control, or optimize the behaviour of the system within these predefined boundaries.

A substantial part of ML-assisted microfluidic design relies on surrogate models developed to approximate CFD simulations. DNNs and other ML models have been used to predict quantities such

as mixing index of micromixers [39], velocity and concentration fields in micromixers [40], flow patterns in cyclone separators [41], and concentration distributions in concentration gradient generators [42], allowing candidate designs to be evaluated much faster than by repeated numerical simulation. However, developing such surrogate models typically requires large training datasets, often ranging from thousands to tens of thousands of CFD simulations. These surrogate models have subsequently been combined with optimization methods such as genetic algorithms for micromixer optimization [43], coordinate-descent approaches for droplet generators [44], simulated annealing for fractal micromixers [45], reinforcement learning for exploring the design space of micromixers [46], and multi-objective evolutionary algorithms [47–49] to balance mixing efficiency, pressure drop, energy consumption, and pumping power. Namely, surrogate models approximate expensive CFD evaluations by learning the relationship between design parameters and device performance. Once trained, they can rapidly predict quantities such as mixing efficiency, particle trajectories, pressure drop, or flow fields for many candidate designs. However, they remain limited to the physical phenomena represented in their training data and selected outputs, and may therefore overlook effects such as local shear, recirculation, or particle accumulation on the channel walls. Therefore, surrogate-based optimization introduces a trade-off between computational efficiency and physical reliability. Specifically, prediction errors may propagate into the optimization process, while generating sufficiently representative training datasets can itself require substantial computational effort.

Among surrogate-based optimization methods, Bayesian optimization [50] is particularly attractive when numerical or experimental evaluations are expensive. By combining a probabilistic surrogate model commonly, a Gaussian process [51] with an acquisition function, Bayesian optimization balances exploration of uncertain regions with exploitation of designs that are predicted to perform well. Unlike approaches that require a large dataset to train a surrogate model before optimization, Bayesian optimization updates its surrogate iteratively as new evaluations become available, allowing the search to progress with relatively few simulations or experiments. The probabilistic nature of the Gaussian process also provides an estimate of prediction uncertainty, which is explicitly used to guide the selection of subsequent evaluations. This approach has been applied to optimization of micromixers and particle separator geometries [52,53], and optimization of operating and geometric parameters in droplet-generation systems [54,55].

Although the search is adaptive, the design variables, objectives, and constraints are still defined in advance by the researcher. This becomes particularly restrictive when unexpected physical behaviour emerges during optimization. For example, a micromixer optimized for mixing efficiency may develop high-shear or stagnant regions, while a droplet generator optimized for droplet size may enter an unstable breakup regime. In conventional workflows, recognizing such effects and reformulating the optimization problem typically requires human intervention. Agentic artificial intelligence (AI) could potentially extend this process by identifying relevant physical evidence, introducing additional metrics or constraints, and adapting the subsequent design strategy. In this perspective, we examine how such adaptive AI approaches could extend current microfluidic design optimization workflows, Figure 1.

In agentic design, AI is not limited to performing a predefined prediction or optimization task, but can potentially modify the workflow, design, or problem formulation in response to intermediate physical results. Such higher-level adaptation may not only improve design optimization but also support the discovery of previously unrecognized physical relationships, limitations, or design principles. Agentic AI systems, with large language models (LLMs) at their core, are still at an early stage of application in microfluidic design. Reference [56] provides an example of LLM-assisted iterative conceptual design optimization by demonstrating the use of GPT-4 to iteratively generate and refine CAD designs of microfluidic components through OpenSCAD. In contrast, reference [57] introduced DropMicroFluidAgents, a multi-agent framework in which LLM-based agents autonomously perform, and coordinate tasks related to droplet microfluidic research. These studies represent early steps toward more autonomous AI-assisted microfluidic workflows, although the broader application of AI agents to iterative, physics-guided microfluidic design remains largely unexplored.

In this perspective, we examine how AI agents could extend microfluidic design optimization beyond fixed search spaces toward adaptive, physics-grounded workflows. We organize emerging approaches according to the level and role of agent involvement, and discuss the evolving role of human

supervision, key technical challenges, and a roadmap toward increasingly autonomous microfluidic design.

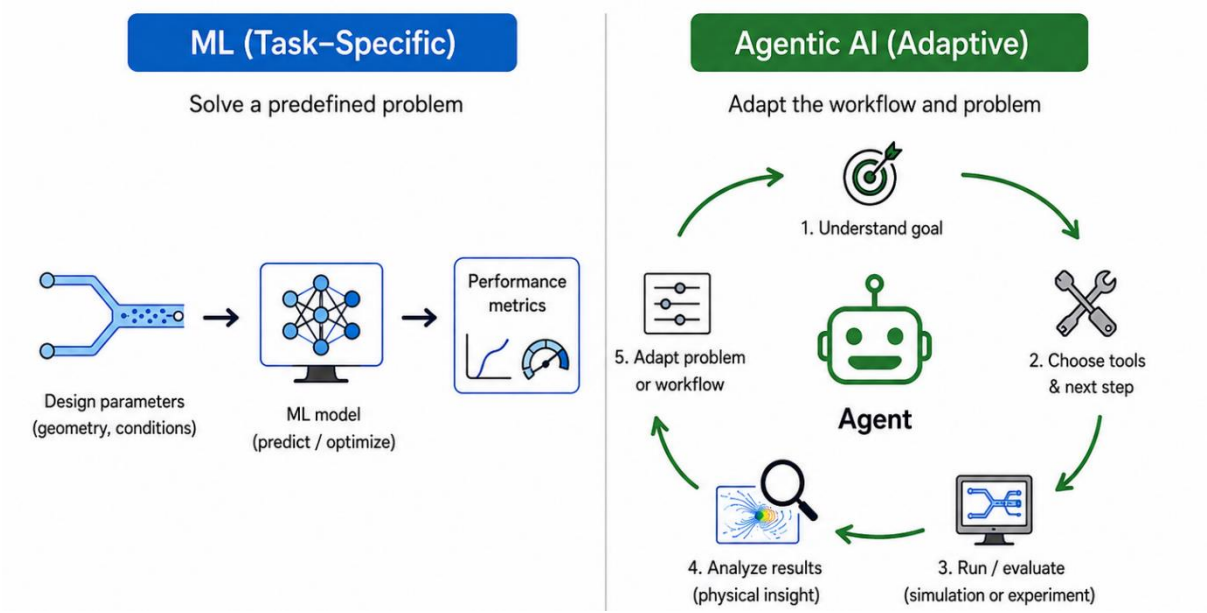


Figure 1. Conceptual distinction between conventional ML-assisted and agentic AI approaches for microfluidic design.

2. Agentic AI Basics

An agentic AI refers to AI autonomous systems that pursue a defined objective through a sequence of decisions and actions. LLMs are increasingly used as the decision-making component of many contemporary agentic systems, particularly when natural-language reasoning, tool coordination, and flexible task decomposition are required. However, agentic systems more broadly do not necessarily rely on LLMs. This perspective focuses specifically on LLM-based agents in the context of microfluidic design.

In this perspective, we use the term LLM-based agent for a system in which the language model has some degree of decision authority over a multi-step workflow. This includes maintaining a task goal, selecting or coordinating tools, interpreting intermediate feedback, and deciding subsequent actions without requiring the user to prescribe every individual step. The degree of agentic behavior therefore depends not only on iteration or tool use, but also on who controls the next action, what information is used as feedback, and how much human intervention is required. A human-directed sequence of LLM interactions, in which the user repeatedly prompts the model and determines each next step, is therefore better described as interactive assistance rather than autonomous agentic control. Likewise, an automated feedback loop may execute repeatedly without being agentic if its sequence of operations and decision rules are fully predefined. A conventional language-model application typically receives an input and generates an output in one interaction. Herein, we distinguish agentic systems from both interactive assistance and conventional automated feedback loops. Interactive assistance remains user-driven, with the model responding to individual requests, while an automated feedback loop repeatedly executes a predefined sequence of operations. By contrast, an agentic system maintains a goal across multiple steps, selects and coordinates actions or tools, evaluates intermediate results, and adapts the workflow accordingly. The defining feature is therefore not simply iteration or automation, but the ability to modify the course of the workflow in response to new information. In microfluidic design, this may include revising design variables, objectives, constraints, or the optimization strategy when intermediate results indicate that the original problem formulation is no longer adequate. The process continues until a stopping condition is reached, such as satisfying a design requirement, exhausting a computational budget, or requiring human approval. An agentic system commonly consists of several interacting components, presented in Figure 2 [58].

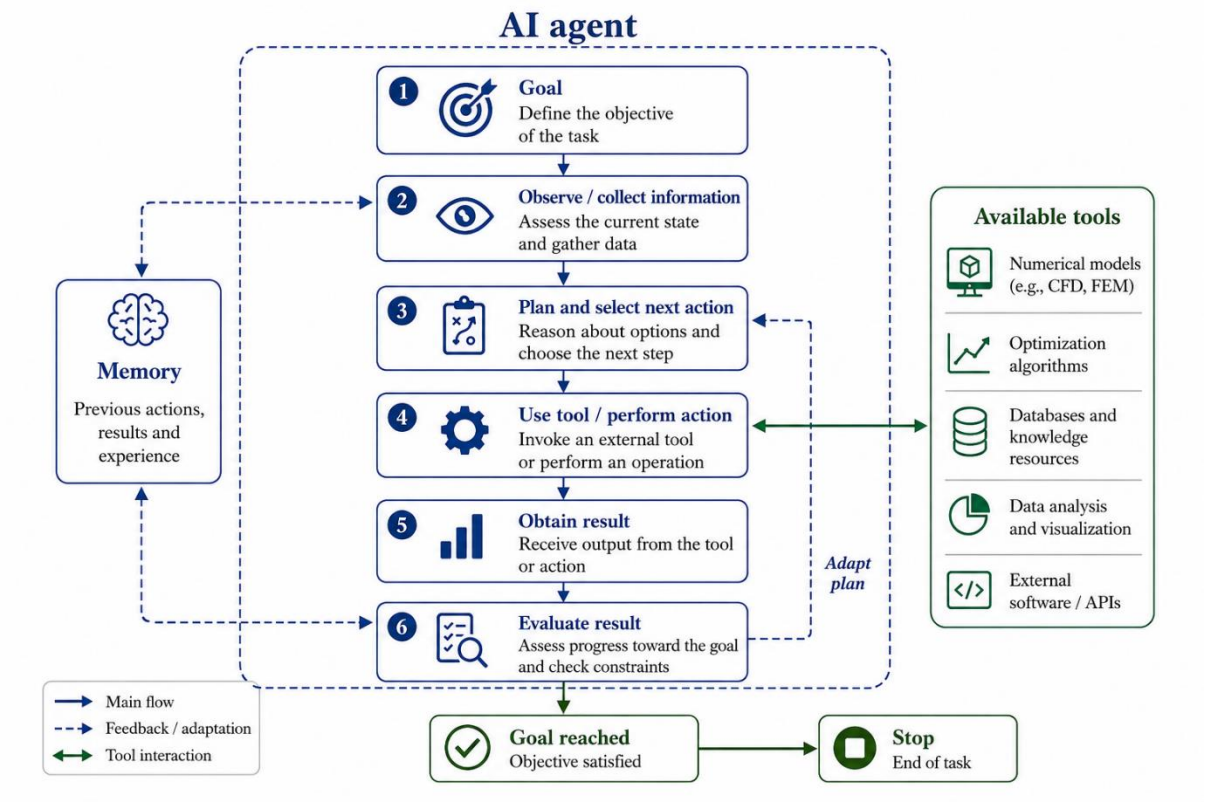


Figure 2. Components of an AI agent.

Namely, the process starts from a defined goal, which specifies the objective of the task. The agent then observes or collects information about the current state of the problem and uses this information to plan and select the next action [59]. This action may involve the use of external tools, such as numerical models, optimization algorithms, databases, data analysis methods, or other software resources [60]. After the action is performed, the agent obtains a result and evaluates whether this result satisfies the objective and relevant constraints. If the goal has been reached, the task is stopped. If not, the agent can adapt the plan and continue with another action. Memory supports this cycle by storing previous actions, results, and experience, allowing the agent to use information from earlier steps when making subsequent decisions [61].

The extent to which these steps are performed autonomously can vary substantially. At a low level of autonomy, the agent may only suggest the next action, while the user remains responsible for execution and final decisions. At a higher level, the agent may independently call simulation software, modify design parameters, compare alternative solutions, evaluate the outcomes, and repeat the process. Therefore, agentic AI does not necessarily imply full autonomy. Rather, it describes a mode of operation in which the system can coordinate and adapt a multi-step workflow in response to the goal and the results obtained along the way [59].

2.1. Large language models as the core of agentic AI

LLMs are commonly used as the central decision-making component of agentic systems. They are trained on large collections of text and other structured representations and learn statistical relationships between words, concepts, instructions, and problem-solving patterns. During use, an LLM generates an output based on the provided prompt, the available context, and the sequence of previously generated tokens [62,63].

The main strength of an LLM is its ability to process information expressed in natural language. This allows it to interpret user requirements, summarize scientific knowledge, compare alternative approaches, translate qualitative goals into technical tasks, and produce instructions for external tools. The same model can therefore operate across several stages of an engineering workflow without

requiring a separate manually programmed rule for each possible situation. LLMs can also generate structured outputs such as parameter lists, simulation configurations, optimization objectives, code, or tool calls [60]. In an agentic system, these outputs are not necessarily the final result. They may represent actions that are executed by other components. For example, an LLM may generate a CFD configuration, request a simulation, inspect the returned performance metrics, and then decide which design should be evaluated next [59]. However, LLMs do not directly solve the governing physical equations of a microfluidic system. Their numerical reasoning may be unreliable, and their outputs may contain unsupported assumptions or physically inconsistent recommendations [60,64]. Their internal knowledge may also be incomplete or outdated [65,66]. For this reason, the LLM should generally be treated as a reasoning and orchestration component rather than as a replacement for simulation, optimization, or experimental validation. Moreover, the behaviour of an LLM is strongly influenced by its context. Relevant information may include the design objective, previous candidate designs, simulation results, fabrication constraints, retrieved scientific knowledge, and instructions defining the available tools. The quality and organization of this context are therefore central to the performance of an agentic system [67–69].

2.2. Core agentic capabilities

Agentic AI systems extend the capabilities of LLMs beyond generating isolated responses. They can pursue an objective through a sequence of actions, interact with external tools, retain relevant information, and adapt their behaviour based on obtained results. Although many individual capabilities can be distinguished, most agentic systems rely primarily on four closely related elements: planning, tool use, memory, and reflection [58,70].

2.2.1. Planning and task decomposition

Planning enables an agent to divide a complex objective into a sequence of smaller and more manageable tasks. Instead of attempting to solve the entire problem in a single response, the agent determines which steps are required, in what order they should be performed, and which information or tools are needed at each stage. A plan may include information retrieval, data preparation, model execution, result evaluation, and revision of previous decisions. Plans may be generated at the beginning of a task or updated dynamically as new information becomes available. The ability to revise a plan is important because initial assumptions may prove incorrect, tools may fail, or intermediate results may indicate that the selected approach is unsuitable. Planning therefore provides structure to the workflow, while replanning enables adaptation [59,71].

2.2.2. Tool use and workflow orchestration

Tool use allows an agent to interact with external systems such as databases, search systems, numerical models, simulation software, optimization algorithms, code execution environments, and data-analysis applications. The agent selects an appropriate tool, prepares its input, invokes it, and interprets the returned output [60]. It may also coordinate several tools within the same workflow. For example, the agent may retrieve relevant information, use it to configure a computational model, execute the model, and then process the resulting data. This separation is important because the language model is primarily responsible for reasoning and decision-making, while specialized tools perform tasks requiring reliable numerical calculation, data retrieval, or domain-specific processing. Tool orchestration enables the agent to combine these components into a single adaptive workflow. External knowledge retrieval can also be viewed as a form of tool use. It allows the agent to ground its decisions in documents, databases, or other sources instead of relying entirely on information encoded during model training [65].

2.2.3. Memory and state management

Memory enables an agent to retain relevant information across multiple actions. Without memory, each decision would be based only on the current input, making it difficult to maintain consistency or learn from previous steps. Short-term memory may contain the current objective, active plan, recent tool outputs, and intermediate decisions. Longer-term memory may store previously solved problems, recurring preferences, successful strategies, and earlier failures [61,72]. Memory is also required for tracking the state of an ongoing workflow. The agent must know which actions have already been

performed, which alternatives have been considered, and which results should influence the next decision. However, storing all available information directly in the language-model context is often inefficient. Practical systems may therefore keep detailed information in external storage and retrieve only the parts relevant to the current action [66].

2.2.4. Feedback, reflection, and adaptation

Agentic systems typically operate through an iterative feedback loop. Reflection refers to the evaluation of previous decisions, generated outputs, and tool results. The agent may assess whether an action satisfied the intended objective, whether an obtained result is credible, or whether the current strategy should be modified. Reflection may lead to correction of an output, repetition of a failed step, selection of a different tool, or revision of the plan [72]. It also supports error recovery when external tools return invalid inputs, execution failures, or unexpected results. The ability to process feedback and adapt distinguishes an agentic workflow from a fixed sequence of predefined operations. However, reflection performed by the same language model that generated the original decision is not a guarantee of correctness [73]. Independent validation, explicit constraints, and human oversight may still be required, particularly in technically demanding or high-consequence applications. Together, planning, tool use, memory, and reflection allow an agent to coordinate multi-step processes and modify its behaviour as the task develops. These capabilities form the basis for the different levels of agent involvement described in Section 4.

3. The agentic potential of microfluidic design

Microfluidic design has several characteristics that make it well suited to agentic workflows. First, it is usually organized around a clearly defined functional goal, such as improving mixing, increasing separation efficiency, controlling droplet size, reducing pressure drop, or limiting shear stress. This goal-oriented structure matches the basic operation of an AI agent, which must interpret an objective and decide which action should be performed next.

Second, microfluidic designing process is inherently iterative. Candidate geometries are proposed, evaluated by numerical simulation or experiment, compared against performance criteria, and then modified. This cycle closely resembles the agentic loop of action, feedback, and adaptation. In this context, an AI agent would use the result of each evaluation to decide how the next design step should change.

Third, the process of microfluidic design uses various tools. A single workflow may involve CAD generation, meshing, CFD simulation, extraction of performance metrics, optimization algorithms, surrogate models, data analysis, visualization, and checks of fabrication constraints. Importantly, many established software tools supporting these tasks already provide APIs, scripting capabilities, or other forms of programmatic access, creating a technical basis for their integration into agentic workflows. Table 1 summarizes representative software tools already used across different stages of microfluidic design and highlights their potential roles in future agentic workflows. An AI agent could act as a coordinating layer that selects, connects, and supervises these tools according to the current state of the design process.

A further advantage is that microfluidic simulations provide rich physical feedback. CFD results include velocity and pressure fields, concentration distributions, shear-stress maps, recirculation regions, secondary flows, or particle trajectories. Such information can help explain why a design succeeds or fails. An agent could therefore use simulations not only to rank candidate geometries, but also to guide physically meaningful modifications of the design strategy.

Finally, microfluidic design involves both numerical and semantic decisions. Some decisions concern continuous parameters, such as geometric parameters (i.e., channel width, flow rate, obstacle spacing, curvature radius, etc.). Others concern the choice of design principle, physical mechanism, fabrication method, objective function, or constraint. Conventional optimization methods are effective when the design space has already been defined, but they generally do not decide whether this definition remains appropriate. Agentic workflows may therefore be most useful at the level where problem formulation, tool coordination, physical interpretation, and design-space adaptation are required.

Table 1. Existing computational tools and potential roles in agentic microfluidic design workflows.

Design task	Existing software/tools	Potential role in an agentic workflow
CAD and geometry generation	AutoCAD, FreeCAD, Onshape, LibreCAD, OpenSCAD	Generation, parameterization, and modification of microfluidic geometries
Meshing	Gmsh, COMSOL Multiphysics, ANSYS Meshing	Automatic mesh generation, refinement, and quality checking
CFD simulation	COMSOL Multiphysics, ANSYS Fluent, OpenFOAM, SimScale	Simulation setup and execution; modification of boundary and operating conditions
Optimization	SciPy, Optuna, Bayesian optimization libraries	Selection of candidate designs and adaptation of the optimization strategy
Data analysis	Python, MATLAB	Extraction of performance metrics and identification of relevant physical features

3.1 When and how should an agent reformulate the design problem?

A central question in agentic microfluidic design is when the current problem formulation should be reconsidered. Reformulation may be needed when intermediate results reveal physical behavior that is not captured by the selected objective, when optimized solutions repeatedly approach fabrication or operating limits, when the search fails to identify feasible designs, or when the assumed model fidelity is insufficient to explain the observed behavior. In such cases, the agent could modify parameter ranges, introduce additional variables or constraints, change the objective or performance metric, switch the optimization strategy or model fidelity, or stop the search and request further numerical or experimental validation. The important distinction is that the agent is not only selecting the next candidate design but also assessing whether the current definition of the design problem remains appropriate.

4. A spectrum of agentic approaches for microfluidic design

The application of agentic AI to microfluidic device design is likely to form a spectrum of methods according to how design authority is distributed among the LLM agent, numerical models, optimization algorithms, domain knowledge, and human experts. At one end of this spectrum, an LLM directly proposes device concepts, geometries, and operating conditions using its internal knowledge and reasoning capabilities. At the other end, the design problem is formulated manually and solved using a fixed numerical optimization workflow, with little or no involvement of an AI agent. Between these extremes are hybrid approaches in which agents interact with literature databases, simulation software, surrogate models, and optimization algorithms.

The position of a method on this spectrum depends primarily on which component performs the main search and decision-making operations. In some systems, the agent directly generates and revises candidate designs. In others, it mainly formulates the optimization problem, configures external tools, or interprets the resulting data. The spectrum therefore reflects not only the amount of agent involvement, but also the nature of that involvement. The approaches are summarized in Figure 3. In practice, agentic systems may integrate elements from different parts of the spectrum, depending on the specific design problem and required level of autonomy. For example, an agent may initially retrieve relevant design principles, use them to define a parameter space for an optimization method, and later analyse CFD results to revise the objective function or constraints. Similarly, different levels of agent autonomy may be appropriate at different stages of the design process.

Physical grounding in Figure 3 concerns how physical knowledge and performance evidence guide design decisions. The ordering follows the increasing scope of search decisions governed by explicit physical representations and numerical rules. LLM-assisted conceptual design draws on learned knowledge, while a knowledge-grounded agent makes the supporting evidence explicit through

traceable sources and physical design rules. Simulation-based search adds numerical evaluation of individual designs. A learned surrogate extends these results into a reusable model linking design variables to physical performance, bringing physical evidence into the screening of prospective candidates. An optimizer carries this structure into candidate selection through explicit objectives, constraints and numerical search rules. In fixed numerical optimization, the formulation and search rules are specified externally. Across these representative configurations, physical information therefore progresses from informing proposals to supporting evaluation and prediction, and finally to governing systematic numerical selection. The configurations can coexist within one workflow.

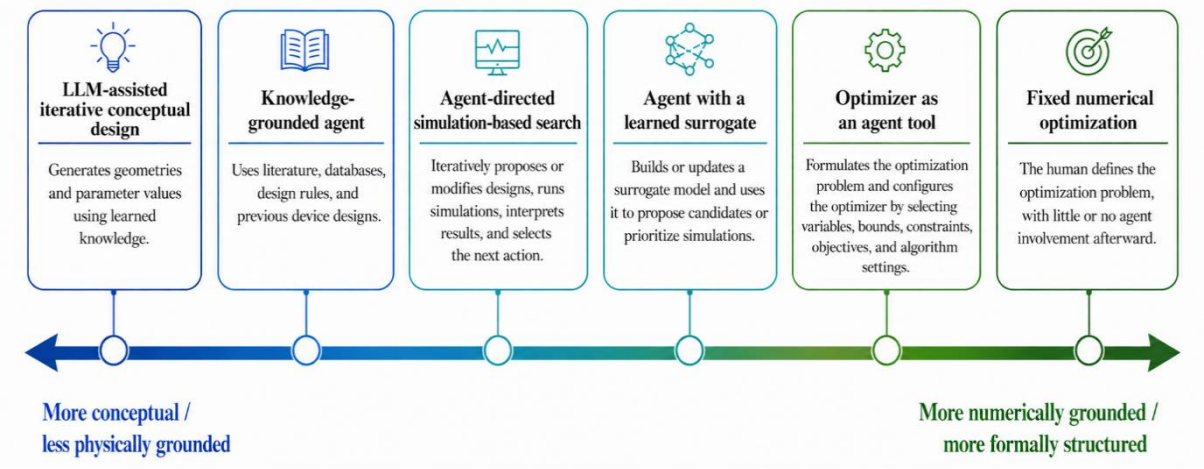


Figure 3. Spectrum of agentic approaches for microfluidic design optimization.

The central part of the spectrum is likely to be particularly important. Pure LLM-based design currently lacks reliable physical and numerical validation, while fully fixed optimization methods are limited by manually selected variables, parameterizations, objectives, and constraints. Hybrid methods can combine the semantic and adaptive capabilities of agents with the numerical reliability of simulation and optimization tools. We anticipate that a substantial part of future research will emerge precisely in this intermediate region, where AI agents progressively assume greater responsibility for problem formulation, tool selection, and iterative decision-making while remaining grounded in physics-based simulation and optimization. In our view, this hybrid regime offers the most realistic pathway toward increasingly autonomous microfluidic design in the coming years.

4.1. LLM-assisted iterative conceptual design

At the conceptual end of the spectrum, an LLM directly generates microfluidic device concepts, geometric configurations, materials, and operating parameters. The design process is primarily based on the knowledge acquired during model training and on the agent's ability to reason about the requested function of the device. Such an approach may be useful during early conceptual development. An agent can rapidly propose alternative channel layouts, mixing mechanisms, droplet-generation structures, separation concepts, or sensing configurations. It can also reason across different levels of abstraction, from functional requirements to possible physical implementations. When implemented as an agentic workflow, this approach requires the LLM to take responsibility for planning and coordinating successive design steps instead of relying on the user to direct each iteration. The agent must decompose the design objective into subproblems, plan a sequence of design decisions, maintain consistency between geometry and operating conditions, and reflect on whether the proposed design satisfies the original requirements. Memory may also be needed to preserve design decisions across multiple iterations. The main advantage of this approach is flexibility. At this conceptual stage, the agent can propose designs before they are encoded in an explicit optimization parameterization. It may propose categorical and structural changes, such as introducing an additional inlet, changing the mixing principle, or replacing a passive mechanism with an active one. However, the absence of quantitative physical feedback is a major limitation. The agent may generate plausible but physically ineffective

designs, violate fabrication constraints, or provide inaccurate numerical values. The resulting concepts should therefore be treated as hypotheses rather than validated designs.

An early example of this approach was demonstrated by Nelson et al. [56], who used GPT-4 to iteratively generate and refine OpenSCAD models of several microfluidic components. The significance of this study is that the LLM was used to translate design instructions into an executable geometric representation that could subsequently be revised through further interaction. Under the definition adopted in Section 2, this implementation is more appropriately classified as interactive LLM assistance, since the iterative design process remains human-directed. Nevertheless, it demonstrates the core capability of LLM-driven geometry generation, which could be extended toward an agentic workflow by allowing the LLM to autonomously plan, evaluate, and revise successive design steps. From a microfluidic perspective, however, the generated geometry remained a conceptual design because its performance was not evaluated. A geometrically valid CAD model may still produce excessive pressure losses, insufficient mixing, or other physically unfavorable behavior. The study therefore illustrates both the potential of LLM-driven geometry generation and the need to connect such systems with physical validation.

4.2. Knowledge-grounded agent

We distinguish knowledge-grounded agents from LLM- assisted iterative conceptual systems by their ability to consult external and updatable information during the design process. Namely, a knowledge-grounded agent extends the LLM-assisted approach by providing access to external sources such as scientific literature, device databases, design guidelines, analytical models, fabrication constraints, and previously tested geometries. The role of the agent is to retrieve relevant information, compare alternative design principles, and synthesize this knowledge into a candidate solution. For example, the agent may identify established relationships between channel dimensions and pressure drop, retrieve known ranges of Reynolds or capillary numbers, or compare fabrication methods before suggesting a design. Planning is needed to determine which information should be retrieved and in what order. Reflection is needed to assess whether the retrieved evidence is consistent, sufficiently relevant, and applicable to the specific design problem. The agent may also need to resolve conflicts between sources or recognize when available information does not cover the requested operating regime. This approach improves traceability because design decisions can be linked to explicit rules, equations, or previously reported devices. It can also reduce obvious physical or fabrication errors and support more informed conceptual design. Nevertheless, basis does not replace direct validation. Published design rules may apply only within specific flow regimes, material systems, or fabrication tolerances. Retrieved information may also be incomplete or biased toward commonly studied device types. The agent must therefore distinguish between evidence-supported design choices and extrapolations beyond available knowledge.

For a micromixer, for example, the agent could retrieve design rules linking channel geometry, Reynolds number, and mixing mechanisms to suggest whether obstacles, curvature, or split-and-recombine structures are appropriate. For droplet generation, it could use reported relationships between capillary number, flow-rate ratio, interfacial tension, and junction geometry to identify a suitable operating regime. Similarly, for particle separation, it could compare inertial, Dean-flow, or deterministic-displacement principles according to particle size, flow conditions, and fabrication limits. In each case, the agent narrows the design choices using existing physical knowledge, but the proposed solution still requires simulation or experimental validation.

4.3. Agent-directed simulation-based search

In agent-directed simulation-based search, the agent iteratively proposes candidate designs, evaluates them using physics-based simulations, interprets the results, and decides how the design should be modified. CFD is the most natural tool for many microfluidic applications, although other models may be used for electrokinetic, thermal, structural, acoustic, or multiphase effects. Unlike fixed optimization, the search strategy is not completely specified in advance. The agent may modify channel dimensions, introduce or remove geometric features, change boundary conditions, or switch between design families based on the observed simulation results. Agentic capabilities are central to this workflow. Planning is

needed to organize the sequence of simulations and decide which aspects of the design should be tested first. Tool use is required to create geometries, generate meshes, configure solvers, run simulations, and extract relevant metrics. Reflection is needed to interpret failed or unexpected results and determine whether the problem lies in the design, the simulation setup, or the evaluation metric. Error recovery is particularly important because numerical simulations may fail to converge or produce nonphysical outputs. The main strength of this approach is that the agent can reason directly from physical feedback. It may inspect not only scalar objective values but also velocity fields, concentration distributions, pressure maps, or local shear stresses. This enables qualitative modifications that are difficult to encode as conventional optimization operators. The main limitation is efficiency. Without a formal search mechanism, the agent may explore the design space inconsistently or repeatedly evaluate similar designs. The process may also be sensitive to prompting and difficult to reproduce. Therefore, agent-directed simulation search is most suitable when the design space contains structural or semantic decisions that cannot easily be represented by a fixed numerical vector.

This type of physics-based feedback is particularly useful when global performance metrics do not reveal why a design succeeds or fails. In micromixers, the agent could inspect concentration and velocity fields and recognize that a high global mixing index still hides poorly mixed or stagnant regions, prompting a structural modification rather than only a parameter adjustment. In droplet generation, multiphase simulations could reveal transitions from stable dripping to jetting or irregular breakup, leading to changes in junction geometry or flow conditions. Similarly, in particle separation, particle trajectories and local flow structures could expose wall accumulation, recirculation, or insufficient lateral migration. In each case, the design process is guided by the observed physical behaviour, allowing the agent to respond to effects that may not be captured by a single predefined performance metric.

4.4. Agent with a learned surrogate

A learned surrogate can be introduced to reduce the computational cost of repeated simulations. In this approach, the agent manages or interacts with a model that approximates the relationship between design variables and device performance. The surrogate may be a Gaussian process, neural network, graph-based model, reduced-order model, or another data-driven approximation. It may also consist of multiple models organized in a modular or hierarchical structure. For example, statistical or classification methods could first identify different operating regimes or design classes, after which specialized surrogate models could be applied to each case. Such an approach may improve prediction accuracy when the design space contains qualitatively different physical behaviors. The surrogate can predict scalar objectives, feasibility, or complete physical fields. The agent can use these predictions to screen candidate designs and reserve expensive CFD simulations for the most informative or promising cases. The agent may be responsible for selecting initial training samples, deciding when additional simulations are required, identifying regions in which the surrogate is unreliable, and updating the model as new data become available. Planning and active selection are needed to balance exploration of uncertain regions with exploitation of high-performing regions. Reflection is needed to detect inconsistencies between surrogate predictions and high-fidelity simulation results. This approach can substantially accelerate design exploration, especially when many similar devices or operating conditions must be evaluated. Knowledge acquired from one design problem may also be transferred to related devices. However, surrogate models introduce an additional source of error. The agent may be guided toward unrealistic or out-of-distribution designs for which the surrogate is inaccurate. A reliable workflow should therefore include uncertainty estimation, periodic validation using physics-based simulations, and explicit checks for violations of physical constraints.

The limitations of surrogate models become particularly important when relevant local physical effects are not included in the predicted outputs. For example, a model may accurately predict mixing efficiency, droplet size, or separation efficiency while overlooking stagnant regions, unstable droplet breakup, or particle accumulation near channel walls. By comparing surrogate predictions with selected high-fidelity simulations, an agent could identify such missing effects and decide when additional training data, outputs, or physical constraints are needed. In this way, the agent can help determine not

only which designs should be explored, but also when the surrogate model is no longer sufficient to describe the underlying physics.

Elements of this approach are already visible in DropMicroFluidAgents [57], where an LLM-based automation designer develops ML models for droplet microfluidic design and generates code-based CAD representations. In the spectrum considered here, this represents an important step beyond isolated LLM-assisted design because several computational activities are coordinated within an agentic framework. However, it does not yet constitute the complete physics-grounded surrogate loop considered above, in which the agent continuously compares surrogate predictions with high-fidelity physical evaluations, identifies regions of insufficient model reliability, and reformulates the design problem when new physical behavior emerges. This distinction indicates a natural direction for extending current agentic microfluidic frameworks.

4.5. Optimizer as an agent tool

In this approach, the main numerical search is performed by an established optimization algorithm, while the agent formulates, configures, and supervises the optimization process. The optimizer is treated as one of several tools available to the agent. The agent may select the design variables, define their ranges, specify objectives and constraints, choose an optimization algorithm, and determine the computational budget. It may also decide whether Bayesian optimization, evolutionary algorithms, gradient-based methods, or topology optimization is most appropriate for the current problem. This division of responsibilities is attractive because it assigns different tasks to components suited to them. The agent handles semantic reasoning, problem formulation, and adaptation, while the optimizer performs systematic numerical search. Established optimization algorithms can already adapt sampling, search over continuous, discrete and categorical variables, explore structural alternatives, and select evaluation fidelity. These adaptations remain the optimizer's responsibility when supported by the chosen implementation. The additional supervisory role assigned to the LLM concerns interpreting qualitative requirements and external domain knowledge, selecting additional diagnostics, and coordinating verification before accepting a revised formulation. For example, the agent may translate a qualitative requirement such as achieving strong mixing with limited cell damage into objectives related to concentration uniformity, pressure drop, and shear exposure. Planning is needed to formulate the optimization workflow and coordinate the optimizer with simulation tools. Reflection can be used to evaluate whether the selected variables and objectives adequately represent the intended device function. If the optimization results are unsatisfactory, the agent may revise parameter bounds, introduce additional constraints, or reformulate the objective. The main advantage is that the search retains the efficiency and reproducibility of numerical optimization while becoming more adaptive. The agent can modify the formulation when the initial problem definition proves inadequate. The main risk is that the agent may configure the optimizer incorrectly. Poorly selected bounds, objectives, or constraints can exclude useful designs or direct the search toward irrelevant solutions. The contribution of the agent must therefore be evaluated separately from the performance of the underlying optimizer.

This division can be particularly useful in microfluidic design, where the initially selected performance metrics may not fully describe device behaviour. For a micromixer, an agent could translate the goal of efficient but gentle mixing into objectives involving mixing uniformity, pressure drop, and shear stress. For droplet generation, it could combine target droplet size with generation stability and robustness to flow-rate variations. In particle separation, the agent could add constraints related to pressure drop or wall accumulation if these effects emerge during simulation. The optimization problem could therefore evolve as additional physical limitations become apparent.

4.6. Fixed numerical optimization

At the opposite end of the spectrum is fixed numerical optimization. In this case, a human expert defines the design variables, parameter ranges, objectives, constraints, simulation model, and optimization algorithm before the search begins. The optimization then proceeds with little or no agent involvement. This approach includes established methods such as Bayesian optimization, evolutionary optimization, gradient-based optimization, and topology optimization. Here, "fixed" refers to workflows that retain an externally specified engineering formulation. The contrast drawn earlier with agentic reformulation

concerns this arrangement of responsibilities; numerical sampling, structural choices and model fidelity may still adapt during optimization. It is particularly appropriate when the design space can be expressed clearly as a set of numerical variables and when the performance metrics are well defined. The main advantages are reproducibility, numerical efficiency, and methodological clarity. The search process can be analyzed systematically, and convergence behaviour can often be characterized more clearly than in an agent-driven workflow. However, the quality of the result depends strongly on the initial problem formulation. Restricting a particular workflow to one geometry family can exclude alternative layouts. This limitation depends on the selected representation. For example, topology optimization can explore different connectivities within its prescribed design domain and constraints. Similarly, poorly chosen objectives may lead to designs that are numerically optimal but practically unsuitable. The method also requires substantial manual effort to formulate the problem and interpret the results. Fixed numerical optimization is therefore likely to remain important, particularly for well-defined engineering problems. Agentic methods are more likely to complement than replace it by supporting problem formulation, design-space adaptation, failure diagnosis, and interpretation of trade-offs.

In microfluidics, this approach is well suited to problems in which the relevant physics and performance metrics are already known. A micromixer may, for example, be optimized for mixing efficiency and pressure drop, a droplet generator for droplet size and frequency, or a particle separator for separation efficiency under a pressure constraint. However, if simulations reveal an unexpected effect such as stagnant regions, unstable droplet breakup, excessive shear, or particle accumulation, the optimization problem must normally be reformulated by the researcher.

Fixed numerical optimization therefore remains highly valuable for well-defined design problems, while agentic approaches may complement it by helping to identify when the original variables, objectives, or constraints no longer adequately represent the observed physical behaviour.

5. The human role in agentic microfluidic design and optimization

The most fundamental human role is to define the engineering problem and determine what constitutes an acceptable solution. Human experts specify the intended device function, relevant physical assumptions, objectives, constraints, acceptable trade-offs, and validation criteria. An agent can help formalize these elements, but it cannot independently determine which performance measures or compromises are scientifically and practically meaningful. The human therefore retains final responsibility for the problem formulation and for decisions based on the resulting designs.

Selecting tools and models: Human experts must choose an appropriate combination of simulation method, surrogate model, and optimization algorithm. This selection depends on the relevant physics, design-space structure, required fidelity, computational budget, and available data. In microfluidic design, this includes deciding whether analytical models are sufficient, whether CFD must represent multiphase, thermal, electrokinetic, or biological effects, and whether Bayesian, evolutionary, gradient-based, or topology optimization is appropriate. The agent may recommend or configure tools, but the human must assess whether their assumptions and capabilities match the problem. Although the integration of microfluidics with ML and automated control is increasing, closer collaboration between microfluidics researchers, data scientists, and computer and software engineers will be required for more advanced agentic workflows.

An additional decision concerns whether the agent should rely on a general-purpose LLM with strong reasoning capabilities or on a model fine-tuned for microfluidic or related engineering tasks. General-purpose models may be particularly suitable when the agent must reason across different domains, interpret loosely specified requirements, compare alternative design principles, or coordinate heterogeneous tools, as in conceptual design or agent-supervised optimization. In contrast, domain-specific fine-tuning may improve consistency in tasks that repeatedly require specialized terminology, design rules, simulation configurations, CAD representations, or interpretation of microfluidic phenomena. However, specialization requires suitable training data and may reduce the flexibility that makes general-purpose models attractive for open-ended design problems. The appropriate choice is therefore likely to depend on the role assigned to the agent, the available domain data, and the required

balance between general reasoning and specialized reliability. We expect future research in agentic microfluidic design to explore both directions, rather than converge on a single model type, with human experts remaining responsible for selecting and validating the model against the requirements of the particular workflow.

Managing context: The active context should contain the information required for the current decision, rather than the complete history of the project. Requirements, tool descriptions, selected literature, current design state, and summarized simulation results compete for a finite context window. Even when a model supports a very long context, including all available material increases cost and latency and may make the most relevant information harder to identify. Representative context limits and rough workflow estimates are shown in Table 2.

Table 2. Representative LLM context limits and estimated context requirements for agentic microfluidic design workflows (data from August 2026).

Category	Model or workflow	Approximate size	Implication
Model limit	GPT-5.6 Sol; Claude Opus 5; Gemini 3.1 Pro Preview	≈1.0 million tokens	Large project states can fit within current long-context models, but selective retrieval and structured summaries remain preferable to loading complete simulation data.
Estimated workload	Single design decision	10,000-40,000 tokens	Requirements, current geometry, tool descriptions, selected domain knowledge, and latest simulation results can be represented directly in context.
Estimated workload	Short optimization campaign	30,000-150,000 tokens	Approximately 10-20 evaluations represented by structured summaries.
Estimated workload	Extended campaign and raw artifacts	150,000-800,000 tokens when summarized; raw fine-mesh CFD data can correspond to hundreds of millions or billions of text-equivalent tokens*	Full meshes, fields, and logs should remain external; only metrics, selected outputs, and diagnoses enter context.

**Note: Raw CFD data can exceed LLM context limits by orders of magnitude; for example, a fine mesh with ($2x10^6$) cells and six stored fields may correspond to ~48 million text-equivalent tokens per solution state, so meshes and field data should remain external while only selected metrics and summaries are provided to the LLM.*

Context management therefore requires external storage, selective retrieval, structured records, rolling summaries, and explicit separation between the current working state and the full project archive. For microfluidic optimization, meshes, complete flow fields, image sequences, and solver logs should normally remain outside the LLM context. The agent should receive compact performance metrics, uncertainty estimates, selected visual evidence, and information needed to explain anomalous behavior.

Evaluating outputs and diagnosing failures: Human experts must assess whether the agent's recommendations and tool outputs are physically plausible, numerically reliable, and aligned with the original objective. They must also diagnose failures from the system's behavior, such as repeated unproductive iterations, invalid tool calls, unsuitable boundary conditions, solver non-convergence, surrogate extrapolation, or optimization of a poor proxy objective. This requires access to the agent's actions, intermediate decisions, and tool results rather than only the final design.

Designing guardrails: Humans must define limits within which the agent may operate. Relevant guardrails include permitted parameter ranges, unit and dimensional-consistency checks, geometric and fabrication constraints, mandatory physical validation, limits on simulation or experimental budgets, and approval points before costly or irreversible actions. Guardrails should also specify when the system must stop, retry, switch methods, or request expert intervention. In this domain, they are not only general AI safety mechanisms but also engineering controls that prevent physically invalid, non-manufacturable, or misleadingly optimized designs.

The human role therefore shifts from manually executing every design step toward defining, supervising, and validating an adaptive computational process. Agentic systems can automate substantial parts of exploration and optimization, but domain expertise remains necessary to ensure that the process addresses the correct problem and that its outputs are scientifically defensible.

6. Technical challenges for reliable agentic microfluidic design

Despite their potential, AI agents cannot yet be treated as fully reliable design systems for microfluidics. Several technical challenges must be addressed before agentic workflows can become practical tools for the microfluidics community. The first challenge is physical reliability. Language-based reasoning may generate plausible design suggestions, but these suggestions may still violate flow-regime assumptions, dimensional consistency, fabrication limits, or operating constraints. For this reason, agentic workflows require systematic checks based on physical relations, CFD validation, and explicit engineering constraints. Moreover, physical reliability also depends on the agreement between numerical predictions and experimental performance. The benefits of shifting more design iterations to the computational stage depend strongly on how accurately the numerical model represents the real device. Numerical models used within agentic design workflows should therefore, whenever possible, be experimentally validated within the relevant operating range. Even for validated models, differences between simulated and experimental performance may arise from fabrication tolerances, material properties, surface effects, or variations in operating conditions. In more advanced workflows, experimental measurements could be incorporated directly as feedback into the design loop. The agent could then identify model–experiment discrepancies and use this information to update the model, modify the design space, or adapt the optimization strategy. Such an approach would not replace experimental validation but could reduce the number of repeated fabrication and testing cycles required to reach a robust final design.

A second challenge is the robustness of tool interaction. Microfluidic design optimization may require the agent to interact with CAD tools, meshing software, CFD solvers, optimization libraries, surrogate models, and data-analysis scripts. Each of these steps can fail or return incomplete, noisy, or difficult-to-interpret outputs. Solver non-convergence, poor mesh quality, incorrect boundary conditions, unit inconsistencies, or failed geometry generation may mislead the agent if they are not detected and handled properly. Reliable agentic systems must therefore include mechanisms for error detection, recovery, and verification of intermediate results.

Another challenge is reproducibility. Conventional optimization workflows are usually defined by explicit variables, objectives, constraints, algorithms, and stopping criteria. In contrast, agentic workflows may adapt their sequence of actions based on intermediate results, which can make them more flexible but also more difficult to reproduce. For scientific use, it will be necessary to record not only the final optimized geometry, but also the agent's decisions, tool calls, parameter changes, failed attempts, and reasons for modifying the design strategy.

The effective use of AI in microfluidic design also depends on how well domain-specific data are organized and made accessible. Device geometries, material properties, fabrication procedures, operating conditions, and performance metrics are often distributed across individual publications and laboratories and are reported in different formats. This fragmentation can limit data reuse and make it difficult to develop AI models that generalize across devices and experimental settings. It also limits the reliability of LLM-based information retrieval, since relevant information may be incomplete, inconsistently reported, outdated, difficult to access, or retrieved without sufficient context. In the near term, individual users or laboratories could create smaller, application-specific knowledge bases containing available materials, fabrication methods, equipment capabilities and operating ranges, previously tested geometries, preferred design concepts, and relevant experimental or simulation results. An agent could use this structured local information to constrain the design space and propose solutions that are compatible with the resources available in a particular laboratory. In the longer term, broader community-curated and machine-readable databases would be needed to support more reliable information retrieval, knowledge transfer across laboratories, improved model generalization, and reduced generation of redundant data. Such databases could also provide a structured knowledge source

from which agents could retrieve prior designs, identify relevant operating regimes, and determine which additional simulations or experiments are needed.

Surrogate-based agentic workflows introduce additional concerns. Although learned surrogate models can reduce the number of expensive CFD simulations, they may become unreliable for out-of-distribution geometries or operating conditions. As previously mentioned, if an agent relies too strongly on an inaccurate surrogate, it may be guided towards designs that appear optimal computationally but are physically invalid or experimentally impractical. Uncertainty estimation, periodic high-fidelity simulation, and explicit feasibility checks should therefore be included when surrogate models are used within agentic design.

Finally, the contribution of the agent itself must be evaluated carefully. Improved performance may result from the underlying optimizer, the CFD model, the selected parameterization, or the agent's decisions about how to modify the workflow. Future studies should therefore separate the performance of the agent from the performance of the tools it uses. This will require benchmark problems, transparent reporting, and comparison with well-defined non-agentic baselines.

6.1. Towards reliable physics-grounded adaptation

Physical reliability remains an open challenge for agentic design: successful simulation execution and plausible interpretation do not necessarily establish correctness. As agentic systems develop stronger grounding, error detection and guardrails, microfluidic research could build on these advances while developing checks suited to particular devices and operating regimes. Promising examples from materials simulation include GENIUS, which combines structured domain knowledge with protocol validation and error recovery [74] , and AutoMOOSE, which introduces separate checks against conservation laws and expected physical trends [75] . Such approaches suggest useful directions for development, although their applicability to microfluidic design requires dedicated evaluation.

Consider a passive micromixer in which adding an obstacle appears to improve the outlet mixing index. An agent might attribute this improvement to enhanced transverse advection. Before extending that design strategy, the workflow could check the imposed flow rates, boundary conditions, fluid properties and diffusivity, together with the applicability of the modeling assumptions. Independent analysis routines could assess fluid and solute mass balances and convergence and recompute the mixing index and pressure drop under mesh and, where relevant, time-step refinement. Sensitivity to the transport discretization could help identify artificial mixing caused by numerical diffusion. Quantifying transverse transport from the computed fields would also provide stronger support than visual inspection alone.

If the apparent improvement disappears under refinement, the appropriate adaptation would be to revisit the numerical setup before continuing geometry search. If it persists, further design changes could be investigated while retaining pressure-drop and fabrication constraints and examining sensitivity to uncertain operating conditions. Acceptance criteria and conditions for expert intervention should be defined for the application. This example illustrates how future agents could coordinate verification before acting on a physical interpretation, with experimental validation remaining necessary to establish correspondence between the model and the device.

7. Roadmap and outlook

The development of AI agents for microfluidic design optimization is likely to proceed gradually rather than through an immediate transition to fully autonomous design, Figure 4. In the near term, agentic systems will probably act as design assistants. They may help researchers summarize literature, compare device concepts, organize simulation results, generate initial parameter ranges, prepare CFD configurations, or identify possible reasons for poor design performance. At this stage, the human expert remains responsible for most decisions, while the agent reduces manual effort and improves workflow organization.

In the medium term, more advanced agents may supervise closed-loop design workflows. Such systems could propose candidate geometries, run simulations, extract performance metrics, update surrogate

models, configure optimization algorithms, and suggest modifications of the design space. Human supervision would still be required, especially for approving changes in objectives, constraints, physical assumptions, or fabrication strategies. This stage is likely to be the most relevant for practical microfluidic design optimization because it combines adaptive reasoning with established numerical tools.

In the longer term, agentic systems may support a more fundamental change in how microfluidic devices are designed. Instead of only optimizing predefined variables, agents could help formulate the design problem itself. They may compare different physical mechanisms, introduce new design variables, reformulate objectives, transfer knowledge between related devices, and explain why a particular design strategy should be pursued. In this scenario, the agent would not simply accelerate parameter search but would participate in shaping the design strategy. Figure 4 organizes agent involvement into five stages. Greater delegated control requires demonstrated capabilities and validation appropriate to the tasks assigned. The benchmark questions in Section 7.1 guide evaluation of workflow-level outcomes, efficiency and reliability. The following capabilities and validation requirements complement these benchmarks with checks specific to each stage:

- Human-led design. Capability: The expert defines the problem and directs the workflow. Validation: Expert-reviewed models and independently assessed design feasibility and performance establish the reference baseline.

- AI design assistant. Capability: The agent retrieves and organizes information and supports selected design tasks. Validation: Recommendations are checked for accuracy, source traceability and consistency with requirements, while reductions in human effort are measured.

- Closed-loop optimization. Capability: The agent coordinates simulations and optimization within defined objectives and constraints. Validation: Feasible-design success, physical performance, computational cost and failure handling are assessed across repeated runs.

- Design strategy partner. Capability: The agent proposes changes to variables, objectives or search strategy in response to evidence. Validation: Reformulations require physical justification, and their benefits are assessed independently against the original application requirements using matched comparisons.

- Autonomous design. Capability: The agent coordinates the design workflow within a defined scope and oversight structure. Validation: Broader delegation requires consistent performance across repeated runs and relevant operating conditions, independent physical validation, and reliable stopping and escalation mechanisms.

Progression between stages therefore depends on demonstrated reliability and practical utility within the intended application, with humans retaining final responsibility.

However, the most realistic outlook is not that AI agents will replace human researchers or established simulation and optimization methods. Their value will lie in connecting these components into adaptive, transparent, and physically grounded workflows. Microfluidic design will remain dependent on domain expertise, accurate modelling, experimental validation, and practical fabrication knowledge. The role of AI agents will be to coordinate these elements more efficiently and to support decisions about what should be simulated, optimized, modified, or validated next.

The transition from fixed optimization workflows to agentic design therefore represents a change in the organization of the design process. Current methods mainly answer the question of which design is best within a predefined search space. Future agentic workflows may also help answer whether the selected search space, objective function, constraints, and design strategy remain appropriate. This shift could move microfluidic design optimization from efficient numerical search towards a more adaptive and reasoning-based engineering process.



Figure 4. Roadmap for the progression of AI agents in microfluidic design optimization. Progression is conditional on demonstrated reliability and utility within the intended application. "Outcome Ownership" describes the agent's role in coordinating the delegated workflow, tracking whether agreed design requirements have been met, and documenting outcomes, validation evidence and unresolved limitations. Humans retain final responsibility for accepting designs and decisions based on them.

7.1. Benchmark questions for future studies

Future studies could assess agentic workflows through representative tasks such as constrained micromixer design, stable droplet generation, or particle separation. Three benchmark questions would be particularly useful:

1. How often does the workflow produce a design satisfying the original physical, operating and fabrication requirements, and what final physical performance is achieved?

2. What resources are required, including the number and fidelity of simulations, total computational cost and elapsed time, and the frequency and active time of human intervention?

3. How reliably does the workflow detect and respond to invalid results or unexpected behaviour, recover from failures or request appropriate intervention, and reproduce its performance across repeated runs?

Comparisons should include workflows using the same simulation models and optimizer without agent supervision, and human-supervised workflows, with matched starting information and resource limits. Failed attempts, verification runs and agent overhead should be included in the reported cost. Ablations disabling selected supervisory capabilities could help identify their contribution. Final designs should be assessed independently against common application requirements, even when the agent revises its internal objectives or search strategy. Reporting distributions across repeated runs would help distinguish consistent benefits from isolated successful outcomes.

8. Conclusion

AI agents offer a possible shift in microfluidic design optimization from fixed parameter search towards adaptive, goal-directed design workflows. Future approaches will likely involve different levels and roles of AI-agent involvement, depending on the design problem, available tools, and required degree of autonomy. Their role is not to replace CFD, surrogate models, optimization algorithms, or human expertise, but to coordinate these components and support decisions about what should be simulated, optimized, modified, or validated next. This perspective is particularly relevant for microfluidics, where device performance depends on strongly coupled geometric, operational, and physical parameters. By

combining simulation results, optimization tools, design constraints, and memory of previous iterations, agentic systems could help identify not only which design performs best, but also why a design succeeds or fails and whether the current design strategy remains appropriate. However, the usefulness of AI agents will depend on their physical grounding, transparency, reproducibility, and integration with human supervision. Agentic workflows must include explicit constraints, reliable numerical validation, uncertainty awareness, and clear records of intermediate decisions. With these elements in place, AI agents may become valuable decision layers that make microfluidic design optimization more flexible, interpretable, and efficient.

Acknowledgements

This research has been supported by the Ministry of Science, Technological Development and Innovation (Contract No. GA No. 451-03-33/2026-03/200358 and 451-03-33/2026-03/200385).

Funding

IK, NJ, and VR acknowledge the financial support of the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under Grant Agreement No. 451-03-33/2026-03/200358, while DM acknowledges financial support under Grant Agreement No. 451-03-33/2026-03/200385.

Author contributions

IK: Methodology, Investigation, Data Curation, Visualization, Writing – Original Draft; OK: Methodology, Investigation, Data Curation, Visualization, Writing – Original Draft; NJ: Writing – Review & Editing, Supervision; DM: Writing – Review & Editing, Supervision, VR: Writing – Review & Editing, Supervision.

Disclosure statement

The authors declare no conflict of interest.

Data availability

The data presented in this study are available on request from the corresponding author. Email: ivana.kundacina@biosense.rs

Declaration of generative AI use

The authors used AI-assisted tools for grammar and spell checking of the manuscript. All scientific content, methodology, data analysis, and conclusions are the sole work of the authors.

References

[1] A. Deb, G. Nagesh, T. Harrison, D. Spicer, D.S.-K. Ting, and M.J. Ahamed, “A review of commercial and laboratory-based microfluidic devices based on glass and/or silicon substrates,” RSC Adv. **16**(14), 12663–12681 (2026).

[2] M.I. Hajam, and M.M. Khan, “Microfluidics: a concise review of the history, principles, design, applications, and future outlook,” Biomater. Sci. **12**(2), 218–251 (2024).

[3] X. Xu, L. Cai, S. Liang, Q. Zhang, S. Lin, M. Li, Q. Yang, C. Li, Z. Han, and C. Yang, “Digital microfluidics for biological analysis and applications,” Lab Chip **23**(5), 1169–1191 (2023).

[4] P. Aryal, C. Hefner, B. Martinez, and C.S. Henry, "Microfluidics in environmental analysis: advancements, challenges, and future prospects for rapid and efficient monitoring," Lab Chip **24**(5), 1175–1206 (2024).

[5] G. Gharib, İ. Bütün, Z. Muganlı, G. Kozalak, İ. Namlı, S.S. Sarraf, V.E. Ahmadi, E. Toyran, A.J. van Wijnen, and A. Koşar, "Biomedical Applications of Microfluidic Devices: A Review," Biosensors (Basel). **12**(11), 1023 (2022).

[6] N. Xiang, and Z. Ni, "Microfluidics for Biomedical Applications," Biosensors (Basel). **13**(2), 161 (2023).

[7] S. Regmi, C. Poudel, R. Adhikari, and K.Q. Luo, "Applications of Microfluidics and Organ-on-a-Chip in Cancer Research," Biosensors (Basel). **12**(7), 459 (2022).

[8] Y. Li, K. Sun, Y. Shao, C. Wang, F. Xue, C. Chu, Z. Gu, Z. Chen, and J. Bai, "Next-Generation Approaches for Biomedical Materials Evaluation: Microfluidics and Organ-on-a-Chip Technologies," Adv. Healthc. Mater. **14**(1), (2025).

[9] L. An, Y. Liu, and Y. Liu, "Organ-on-a-Chip Applications in Microfluidic Platforms," Micromachines (Basel). **16**(2), 201 (2025).

[10] S. Surappa, P. Multani, U. Parlatan, P.D. Sinawang, J. Kaifi, D. Akin, and U. Demirci, "Integrated 'lab-on-a-chip' microfluidic systems for isolation, enrichment, and analysis of cancer biomarkers," Lab Chip **23**(13), 2942–2958 (2023).

[11] U.A. Gurkan, D.K. Wood, D. Carranza, L.H. Herbertson, S.L. Diamond, E. Du, S. Guha, J. Di Paola, P.C. Hines, I. Papautsky, S.S. Shevkoplyas, N.J. Sniadecki, V.K. Pamula, P. Sundd, A. Rizwan, P. Qasba, and W.A. Lam, "Next generation microfluidics: fulfilling the promise of lab-on-a-chip technologies," Lab Chip **24**(7), 1867–1874 (2024).

[12] D. McIntyre, A. Lashkaripour, P. Fordyce, and D. Densmore, "Machine learning for microfluidic design and control," Lab Chip **22**(16), 2925–2937 (2022).

[13] M. Ferreira, V. Carvalho, J. Ribeiro, R.A. Lima, S. Teixeira, and D. Pinho, "Advances in Microfluidic Systems and Numerical Modeling in Biomedical Applications: A Review," Micromachines (Basel). **15**(7), 873 (2024).

[14] G. Esposito, S. Romano, M.A. Hulsen, G. D'Avino, and M.M. Villone, "Numerical simulations of cell sorting through inertial microfluidics," Physics of Fluids **34**(7), (2022).

[15] R. Shi, "Numerical simulation of inertial microfluidics: a review," Engineering Applications of Computational Fluid Mechanics **17**(1), (2023).

[16] J. Zhou, J. Dong, H. Hou, L. Huang, and J. Li, "High-throughput microfluidic systems accelerated by artificial intelligence for biomedical applications," Lab Chip **24**(5), 1307–1326 (2024).

[17] J. Park, Y.W. Kim, and H.-J. Jeon, "Machine Learning-Driven Innovations in Microfluidics," Biosensors (Basel). **14**(12), 613 (2024).

[18] J. Lai, S. Gaira, R. Wang, Q. Zhang, M. Li, L. Wu, D. Li, Y. Li, C. Zhang, J. Guo, and S. Tang, "Integrating Artificial Intelligence With Droplet-Based Microfluidics: Advances, Challenges, and Emerging Opportunities," Advanced Intelligent Systems **8**(5), (2026).

[19] F. Del Giudice, "A perspective on machine learning applied to droplet microfluidics," Biomicrofluidics **20**(2), (2026).

[20] P. Hadikhani, N. Borhani, S.M. H. Hashemi, and D. Psaltis, "Learning from droplet flows in microfluidic channels using deep neural networks," Sci. Rep. **9**(1), 8114 (2019).

[21] J.W. Khor, N. Jean, E.S. Luxenberg, S. Ermon, and S.K.Y. Tang, "Using machine learning to discover shape descriptors for predicting emulsion stability in a microfluidic channel," Soft Matter **15**(6), 1361–1372 (2019).

[22] A. Samari, K. Jannati, and A. Jafari, "A data driven framework for optimizing droplet microfluidics with residual block and Fourier enhanced networks," Sci. Rep. **15**(1), 29077 (2025).

[23] A. Arjun, R.R. Ajith, and S. Kumar Ranjith, "Mixing characterization of binary-coalesced droplets in microchannels using deep neural network," Biomicrofluidics **14**(3), (2020).

[24] M.T. Doganay, P. Chakraborty, S.M. Bommakanti, S. Jammalamadaka, D. Battalapalli, A. Madabhushi, and M.S. Draz, "Artificial intelligence performance in testing microfluidics for point-of-care," Lab Chip **24**(21), 4998–5008 (2024).

[25] J. Shi, L. You, Y. Wang, and B. Peng, "Deep learning based method for dynamic tracking of bubbles in microfluidic gas-driven water experiments," Geoenergy Science and Engineering **243**, 213391 (2024).

[26] G. Antonelli, J. Filippi, M. D'Orazio, G. Curci, P. Casti, A. Mencattini, and E. Martinelli, "Integrating machine learning and biosensors in microfluidic devices: A review," Biosens. Bioelectron. **263**, 116632 (2024).

[27] A. Jemmali, S. Kaziz, F. Echouchene, and M.H. Gazzah, "Optimization of Lab-On-a-CD by Experimental Design and Machine Learning Models for Microfluidic Biosensor Application," IEEE Sens. J. **24**(7), 9299–9307 (2024).

[28] I. Ben Romdhane, A. Jemmali, S. Kaziz, F. Echouchene, T. Alshahrani, and H. Belmabrouk, "Taguchi method: artificial neural network approach for the optimization of high-efficiency microfluidic biosensor for COVID-19," The European Physical Journal Plus **138**(4), 359 (2023).

[29] S. Kaziz, A. Jemmali, and F. Echouchene, "Optimization of annular microfluidic biosensor enhanced by active and passive effects using Taguchi's method coupled with multi-layer perceptron neural networks (MLP-NN) models," Microfluid. Nanofluidics **27**(9), 60 (2023).

[30] X. Wang, J. Liu, R. Dong, M.D. Gilchrist, and N. Zhang, "High-precision digital light processing (DLP) printing of microstructures for microfluidics applications based on a machine learning approach," Virtual Phys. Prototyp. **19**(1), (2024).

[31] A. Norouzi, E. Ahmadinejad, I. Cakiroglu, H. Celik, M. Tugrul Birtek, A. Ahmadpour, and S. Tasoglu, "Machine learning-assisted optimization of CNC-micromilled PMMA microchannels for predictable hydraulic performance and low protein fouling resistance," Biosens. Bioelectron. X **30**, 100781 (2026).

[32] K. Nathanael, S. Cheng, N.M. Kovalchuk, R. Arcucci, and M.J.H. Simmons, "Optimization of microfluidic synthesis of silver nanoparticles: A generic approach using machine learning," Chemical Engineering Research and Design **193**, 65–74 (2023).

[33] A.A. Lendhe, N. Raykar, B.S. Kale, and K.S. Bhole, "Machine learning approach to predict viscous fingering in Hele-Shaw cells," International Journal on Interactive Design and Manufacturing (IJIDeM) **18**(10), 7183–7239 (2024).

[34] O.J. Dressler, P.D. Howes, J. Choo, and A.J. deMello, "Reinforcement Learning for Dynamic Microfluidic Control," ACS Omega **3**(8), 10084–10091 (2018).

[35] B. Liu, C. Jiang, Q. Xu, H. Yao, T.-Y. Ho, and B. Yuan, "Efficient Routing-Based Synthesis for Digital Microfluidic Biochips via Reinforcement Learning," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **45**(1), 162–175 (2026).

[36] J. Shi, P. Fu, and W. Zheng, "A design method based on Bayesian decision for routing-based digital microfluidic biochips," Analyst **147**(6), 1076–1085 (2022).

[37] M.T. Birtek, V. Aktas, B. Aktas, A.C. Abdullah, A. Ozcan, and S. Tasoglu, "ML-automated microfluidic circuit design," Sci. Adv. **12**(5), (2026).

[38] M. Khalghollah, A. Zare, S. Abdollahi, B.H. Far, and A. Sanati Nezhad, "Machine Learning-Driven Capillary Microfluidic Design Automation for Programmable Gradient Generation and Antimicrobial Testing," Small **22**(37), (2026).

[39] M.T. Birtek, M.M. Alseed, M.R. Sarabi, A. Ahmadpour, A.K. Yetisen, and S. Tasoglu, "Machine learning-augmented fluid dynamics simulations for micromixer educational module," Biomicrofluidics **17**(4), (2023).

[40] J. Wang, N. Zhang, J. Chen, G. Su, H. Yao, T.-Y. Ho, and L. Sun, "Predicting the fluid behavior of random microfluidic mixers using convolutional neural networks," Lab Chip **21**(2), 296–309 (2021).

[41] L.H. Queiroz, F.P. Santos, J.P. Oliveira, and M.B. Souza, "Physics-Informed deep learning to predict flow fields in cyclone separators," Digital Chemical Engineering **1**, 100002 (2021).

[42] N. Zhang, Z. Liu, and J. Wang, "Machine-Learning-Enabled Design and Manipulation of a Microfluidic Concentration Gradient Generator," Micromachines (Basel). **13**(11), 1810 (2022).

[43] D. de O. Maionchi, L. Ainstein, F.P. dos Santos, and M.B. de Souza Júnior, "Computational fluid dynamics and machine learning as tools for optimization of micromixers geometry," Int. J. Heat Mass Transf. **194**, 123110 (2022).

[44] A. Lashkaripour, C. Rodriguez, N. Mehdipour, R. Mardian, D. McIntyre, L. Ortiz, J. Campbell, and D. Densmore, "Machine learning enables design automation of microfluidic flow-focusing droplet generation," Nat. Commun. **12**(1), 25 (2021).

[45] H. Lv, X. Chen, and X. Zeng, "Optimization of micromixer with Cantor fractal baffle based on simulated annealing algorithm," Chaos Solitons Fractals **148**, 111048 (2021).

[46] Y. Chen, T. Sun, Z. Liu, Y. Zhang, and J. Wang, "Towards Design Automation of Microfluidic Mixers: Leveraging Reinforcement Learning and Artificial Neural Networks," Micromachines (Basel). **15**(7), 901 (2024).

[47] F.-J. Granados-Ortiz, and J. Ortega-Casanova, "Machine learning-aided design optimization of a mechanical micromixer," Physics of Fluids **33**(6), (2021).

[48] J. Wang, N. Zhang, J. Chen, V.G.J. Rodgers, P. Brisk, and W.H. Grover, "Finding the optimal design of a passive microfluidic mixer," Lab Chip **19**(21), 3618–3627 (2019).

[49] H. Lv, X. Chen, X. Li, Y. Ma, and D. Zhang, "Finding the optimal design of a Cantor fractal-based AC electric micromixer with film heating sheet by a three-objective optimization approach," International Communications in Heat and Mass Transfer **131**, 105867 (2022).

[50] D.R. Jones, M. Schonlau, and W.J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," Journal of Global Optimization **13**(4), 455–492 (1998).

[51] C.E. Rasmussen, "Gaussian Processes in Machine Learning," (2004), pp. 63–71.

[52] I. Kundacina, O. Kundacina, D. Miskovic, and V. Radonic, "Advancing microfluidic design with machine learning: a Bayesian optimization approach," Lab Chip **25**(4), 657–672 (2025).

[53] N. Oluški, O. Kundačina, V. Radonić, M. Bukurov, and I. Kundačina, "Constrained Bayesian optimization-driven design of a microfluidic particle separator under a pressure-drop limit," Results in Engineering **32**, 112986 (2026).

[54] Z. Li, X. Guan, J. Zhang, Q. Guo, Q. Xu, and N. Yang, "Bayesian optimization of operational and geometric parameters of microchannels for targeted droplet generation," Chin. J. Chem. Eng. **84**, 244–253 (2025).

[55] S. Cho, H. Kim, S. Shin, M. Lee, and J. Lee, "Autonomous Bayesian Optimization-Based Control System for Droplet Generation," Small Methods **9**(11), (2025).

[56] M.D. Nelson, B.L. Goenner, and B.K. Gale, "Utilizing ChatGPT to assist CAD design for microfluidic devices," Lab Chip **23**(17), 3778–3784 (2023).

[57] D.-N. Nguyen, R.K.-Y. Tong, and N.-D. Dinh, "DropMicroFluidAgents (DMFAs): autonomous droplet microfluidic research framework through large language model agents," Digital Discovery **4**(10), 2827–2851 (2025).

[58] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W.X. Zhao, Z. Wei, and J. Wen, "A survey on large language model based autonomous agents," Front. Comput. Sci. **18**(6), 186345 (2024).

[59] Yao Shunyu, Zhao Jeffrey, Yu Dian, Du Nan, Shafran Izhak, and Narasimhan Karthik, "ReAct: Synergizing Reasoning and Acting in Language Models," in *International Conference on Learning Representations (ICLR)*, (2023).

[60] T. Schick, J. Dwivedi-Yu, R. Dessi, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language Models Can Teach Themselves to Use Tools," in *Advances in Neural Information Processing Systems 36*, (Neural Information Processing Systems Foundation, Inc. (NeurIPS), San Diego, California, USA, 2023), pp. 68539–68551.

[61] J.S. Park, J. O'Brien, C.J. Cai, M.R. Morris, P. Liang, and M.S. Bernstein, "Generative Agents: Interactive Simulacra of Human Behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, (ACM, New York, NY, USA, 2023), pp. 1–22.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, (2017).

[63] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, (2020).

[64] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring How Models Mimic Human Falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Association for Computational Linguistics, Stroudsburg, PA, USA, 2022), pp. 3214–3252.

[65] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, (2020).

[66] N.F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the Middle: How Language Models Use Long Contexts," Trans. Assoc. Comput. Linguist. **12**, 157–173 (2024).

[67] R. Asgari, M. Moradi, K. Yan, D. Colwell, and M. Samwald, "A Critical Review of Methods and Challenges in Large Language Models," Computers, Materials & Continua **82**(2), 1681–1698 (2025).

[68] M.C. Ramos, C.J. Collison, and A.D. White, "A review of large language models and autonomous agents in chemistry," Chem. Sci. **16**(6), 2514–2572 (2025).

[69] M.A.K. Raiaan, Md.S.H. Mukta, K. Fatema, N.M. Fahad, S. Sakib, M.M.J. Mim, J. Ahmad, M.E. Ali, and S. Azam, "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," IEEE Access **12**, 26839–26874 (2024).

[70] X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen, "Understanding the planning of LLM agents: A survey," (2024).

[71] S. Yao, D. Yu, J. Zhao, I. Shafran, T.L. Griffiths, Y. Cao, and K. Narasimhan, "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, (2023).

[72] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, (2023).

[73] J. Huang, X. Chen, S. Mishra, H.S. Zheng, A.W. Yu, X. Song, and D. Zhou, "Large Language Models Cannot Self-Correct Reasoning Yet," in *International Conference on Learning Representations (ICLR 2024)*, (2024).

[74] M. Soleymanibrojeni, R. Aydin, D. Guedes-Sobrinho, A.C. Dias, M.J. Piotrowski, W. Wenzel, and C.R.C. Rêgo, "GENIUS: an agentic AI framework for autonomous design and execution of simulation protocols," Commun. Mater. **7**(1), 115 (2026).

[75] S. Manna, H. Chan, and S.K.R.S. Sankaranarayanan, "AutoMOOSE: An Agentic AI for Autonomous Phase-Field Simulation," (2026).