

Document Version

Final published version

Licence

CC BY

Citation (APA)

Rupprecht, S., Gao, Q., Karia, T., & Schweidtmann, A. M. (2026). Multi-agent systems for chemical engineering: a review and perspective. *Current Opinion in Chemical Engineering*, 51, Article 101209. <https://doi.org/10.1016/j.coche.2025.101209>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership. Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Review

Multi-agent systems for chemical engineering: a review and perspective

Sophia Rupprecht, Qinghe Gao, Tanuj Karia and Artur M Schweidtmann



Large language model (LLM)-based multi-agent systems (MASs) are a recent but rapidly evolving technology with the potential to transform chemical engineering by decomposing complex workflows into teams of collaborative agents with specialized knowledge and tools. This review surveys the state-of-the-art of MASs within chemical engineering. While early studies demonstrate promising results, scientific challenges remain, including the design of tailored architectures, integration of heterogeneous data modalities, development of foundation models with domain-specific modalities, and strategies for ensuring transparency, safety, and environmental impact. As a young but fast-moving field, MASs offer exciting opportunities to rethink chemical engineering workflows.

Address

Delft University of Technology, Department of Chemical Engineering, Van der Maasweg 9, Delft 2629 Hz, the Netherlands

Corresponding author: Schweidtmann, Artur M (a.schweidtmann@tudelft.nl).

Current Opinion in Chemical Engineering 2026, **51**:101209

This review comes from a themed issue on **Artificial intelligence and chemical engineering**

Edited by **Venkat Venkatasubramanian** and **Connor Coley**

Available online xxxx

<https://doi.org/10.1016/j.coche.2025.101209>

2211–3398/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Introduction

The advent of large language models (LLMs), such as the Generative Pretrained Transformer (GPT-4) [1], has marked a technological leap, enabling advanced capabilities in language translation, multimodal data processing, mathematical reasoning, and code generation [2]. These advances have facilitated progress in various domains, including chemistry and materials science. Likewise, there is growing interest in developing and applying LLMs in the chemical engineering domain [3,4] and more specifically to process systems engineering [5–7].

Although LLMs have demonstrated strong capabilities in various domains, their application in chemical engineering remains limited due to multiple fundamental shortcomings. Chemical engineering workflows are inherently complex, spanning multiple scales and relying heavily on specialized simulation tools and external data sources. Such workflows cannot be reliably replicated or replaced by standalone LLMs yet. Trained as end-to-end black box models, LLMs are prone to hallucinations arising from biases and gaps in their training data, producing confident yet incorrect outputs. Employing such standalone LLMs lacks transparency and interpretability, and can even lead to significant safety risks. Critically, general-purpose LLMs do not possess an intrinsic understanding of physical laws, such as thermodynamic constraints or mass and energy balances, frequently resulting in outputs that are physically invalid [8,9]. Thereby, LLMs often fail to solve domain-specific, multi-step problems that demand physical reasoning and a system-level perspective. Without access to relevant tools and trusted data sources, even the most advanced LLMs remain fundamentally constrained in their capabilities.

LLM-based multi-agent systems (MASs) offer a promising methodology to address these challenges. An agent is most generally defined as an entity that perceives its environment and independently executes actions upon the environment on behalf of its owner [10,11]. In contrast to rule-based agentic frameworks, an LLM-based agent is a dynamic computational entity that is capable of communicating through natural language and can attempt to reason over a broader range of tasks [12]. In the context of MASs, LLM-based agents address the shortcomings of standalone LLMs by extending their capabilities to interact with tools and databases, enabling them to access external knowledge for more informed decision-making and increasing the transparency and traceability of information. MASs elicit inter-agent cooperation, coordination, and negotiation to accomplish individual or shared goals [11]. MASs subdivide complex tasks into smaller sub-tasks and assign roles to distinct agents that can interact with each other [13,14]. Therein, single agents are usually LLMs that are equipped with carefully selected external knowledge in the form of executable tools, access to databases, and literature to perform specific tasks [2]. The individual

agents need to possess a certain level of autonomy, decision-making capabilities, proactiveness, and communication skills to make the MAS performative in dynamic and distributed environments.

MASs have already shown success in other domains [15,16]. Notable examples of MASs are in chemistry [17–19••], drug discovery [20], material design [21], scientific discovery [22–24], medicine [25], and code generation [26]. Recent production-ready frameworks such as Clint and CodeRabbit further showcase the use of multi-agent collaboration in software engineering tasks, including autonomous development, code review, and project management. Recently, several review and perspective articles have specifically focused on the potential of MASs in subfields of chemical engineering [9,27,28]. However, a comprehensive MAS for chemical engineering has not been proposed or developed yet.

In this article, we review the developments of MASs within the overarching chemical engineering domain and provide an outlook for further development and applications. In *State of the art*, we review the recent studies applying tailored, LLM-supported MASs demonstrating the potential of MASs for chemical engineering applications. Furthermore, in *Perspective*, we discuss the fundamental challenges that must be addressed to enable cross-scale integration in chemical engineering workflows with MASs. Finally, we conclude the paper in *Conclusions*.

State of the art

In this section, we first introduce the core concepts of MASs (*Multi-agent systems*) and then review literature on applications of MASs in chemical engineering (*Multi-agent systems in chemical engineering*).

Multi-agent systems

MASs are distributed systems composed of multiple agents that interact to achieve individual or collective goals [2]. While traditional supervised models learn static mappings, agents can dynamically reason over tasks, invoking tools or querying data as needed. These capabilities facilitate iterative and reflective interaction in agents, thereby improving transparency and adaptability compared to standalone LLMs. Human-in-the-loop guidance is required to ensure the alignment of the MAS conversation with the overall objective of the human user. A human-in-the-loop should also be in place to address shortcomings of agents' outputs during conversation. Hence, MASs with human-in-the-loop provide a natural framework for streamlining chemical engineering workflows that require complex decision-making. While MASs were proposed back in the 1990s for autonomous agents and distributed AI [11], they

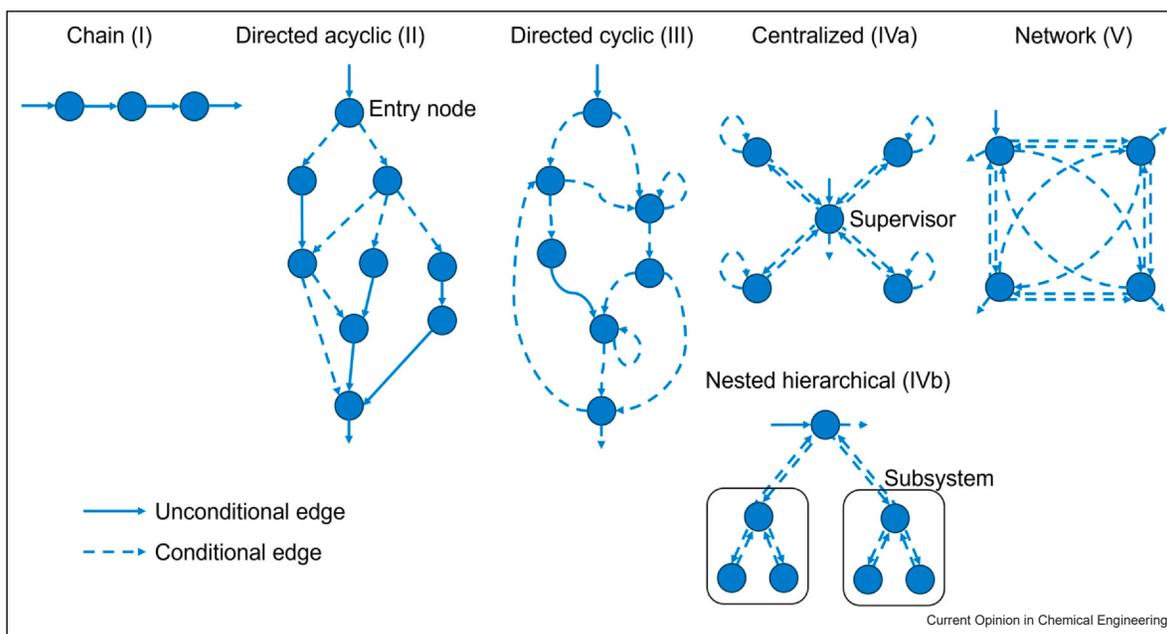
have received significant attention recently due to the advent of LLMs.

For modern MASs, agents are typically LLMs capable of interacting with tools by selecting suitable tools, providing tools' input arguments, and interpreting tool outputs. Agents are also capable of querying structured information from relational databases (e.g. via Structured Query Language (SQL) [29]) and knowledge graphs (e.g. via Cypher [30]), enabling access to structured domain knowledge. A knowledge graph is a structured network of entities (nodes) and their relationships (edges) that encodes meaning and context, enabling both humans and machines to query, integrate, and reason over connected data. For unstructured information — such as scientific literature, reports, or manuals — retrieval-augmented generation (RAG) [31] techniques are used to extract relevant content. Large unstructured text corpora are segmented into smaller text chunks during preprocessing. Each chunk is then transformed into a semantically rich vector representation, or embedding, and stored in a vector database. RAG finally allows the retrieval of semantically similar text snippets during inference, given a user query. Recent extensions adapt RAG for graph-based retrieval, that is, Graph RAG [32–34], by transforming unstructured text into knowledge graphs. The knowledge graphs used for Graph RAG-based approaches offer the advantage of capturing top-level relations between pieces of information, compared to isolated text snippets in the case of standard RAG methods [32].

To facilitate communication among various agents to develop MASs, several open-source libraries such as AutoGen [13••], crewAI [35] by Microsoft, and LangGraph [36] by LangChain have been proposed. There exist many more open-source MAS frameworks, such as CAMEL [37] and MetaGPT [38]. Such libraries offer toolkits to design MASs and infrastructure to use a variety of LLMs as drop-in models within agents. For instance, LangChain supports interfaces for integrating LLMs from multiple providers, including OpenAI, Ollama, and Anthropic. MASs, of course, cannot eliminate the hallucinations of the LLMs they use within agents. For instance, Taskiran et al. [39•] explored a LangChain prebuilt chatbot for pharmaceutical engineering and observed significant propagation of hallucinations in the case of knowledge graph information retrieval.

LangGraph [36] is widely used for scientific applications and supports flexible conversation via graph structures. In LangGraph's graph structure, nodes represent agents, and edges represent communication. LangGraph [36] is also capable of differentiating between unconditional and conditional edges. Unconditional edges imply that there is a predetermined next agent, independent of the contribution of the previous agent. Conditional edges indicate that the next agent is dynamically inferred

Figure 1



Architecture options for MASs. MASs can be modeled as graphs [36] with agents as nodes and edges as conversation routes. Unconditional edges imply that there is a predetermined next agent, independent of the contribution of the previous agent. Conditional edges indicate that the next agent is dynamically inferred during inference, depending on the state of the conversation. The chain architecture (I) is a fixed sequence of agents connected with conditional edges. A directed acyclic conversation graph (II) contains conditional edges. A directed cyclic conversation graph (III) includes loops where substeps of the conversation can be repeated. In a centralized architecture (IVa), there is a single supervisor agent managing multiple subordinate agents. A nested hierarchical structure (IVb) includes a supervisor managing multiple sub-systems, which can consist of single agents or independent, customized architectures. The network architecture (V) puts each agent in direct contact with every other agent independently.

during inference, depending on the state of the conversation. These frameworks can incorporate human agents, enabling expert oversight and intervention, enhancing reliability, interpretability, and control - crucial for safety-critical domains like chemical engineering.

Figure 1 illustrates five basic architecture types of MASs [15,16,36,40]: (I) chain — a linear sequence of agents using unconditional edges only; (II) directed acyclic graph (DAG) — incorporates dynamic edges to enable branching paths; (III) directed cyclic graph (DCG) — allows iterative workflows by including cycles in the graph; (IVa) centralized — a supervisor agent coordinates specialized sub-agents; (IVb) nested hierarchy — an extension of the centralized model where the supervisor manages modular sub-systems, each with its own internal structure; (V) network — fully connected agents that communicate freely and can independently terminate or respond to tasks. The architectures I to V are organized with increasing inter-agent connectivity, which can increase the flexibility of the MAS's behavior. Such increased flexibility could allow for more feedback mechanisms between agents and humans-in-the-loop, and can potentially lead to an increased level of autonomy of the MAS. The choice of architecture depends

on the nature of the application, including task complexity, communication needs, and coordination overhead. Each architecture offers trade-offs in terms of scalability, interpretability, and adaptability.

Multi-agent systems in chemical engineering

In Table 1, we summarize recent studies showcasing the application of MASs for chemical engineering tasks [39••,41•–49••]. Lee et al. [41•] propose a MAS that suggests process design improvements based on existing Process Flow Diagrams (PFDs). Pajak et al. [45] use a MAS to balance economic and environmental objectives in gas-oil separation. MASs have also been employed for general process problem-solving [44], autonomous industrial control [42], and generating PFDs or P&IDs from natural language descriptions [43•46•]. Others use agents to identify operational constraints in process optimization [47] or to build structured knowledge graphs from multimodal data for schematic synthesis via GraphRAG [48]. Approaches to MAS applications in quantum chemistry [49••] and pharmaceutical engineering [39•] are included in the review as well. Now, we examine the publications along five aspects: architecture of the MASs, tool and database integration, modalities of data used, and human integration.

Table 1
Overview of MASs in the chemical engineering domain. The existing literature is categorized by architecture type, task addressed, examples of tool and database integration, human integration, and framework used for implementation.

Literature	Task	MAS architecture	Tools	Database integration	Human integration	Framework
Lee et al.[41•]	Process improvement	Chain	ChemicalAid, Literature data extraction	-	-	OpenAI
Vyas et al.[42]	Industrial control	DCG	Arduino temperature control lab	-	-	CrewAI
Gowaikar et al.[43•]	P&ID generation from text	DCG	MS Visio	-	✓	n/a
Sakhinana et al.[44]	General operations assistance	Centralized	Stack Overflow, Wolfram Alpha	Graph RAG	-	n/a
Pajak et al.[45]	Operational decision-making	DCG	Pyomo, Aspen HYSYS	RAG	-	LangGraph
Srinivas et al.[46•]	PFD and P&ID generation	DCG	-	RAG& Graph RAG	✓	n/a
Zeng et al.[47]	Process optimization	DCG	IDAES simulation	-	-	AutoGen
Srinivas et al.[48]	PFD and P&ID generation	DCG	Web search, Multimodal information extraction	Graph RAG	✓	n/a
Zou et al.[49••]	Computational quantum chemistry task support	Nested hierarchical	DFT calculations RDKit, ORCA, xTB	-	✓	LangGraph & LangChain
Taskiran et al.[39•]	Pharmaceutical engineering chatbot	Chain	-	Knowledge graph	-	LangChain

Architecture

The DCG (III) is the most common architecture type of MASs in chemical engineering [42,43,45–47] (Table 1). MASs using a DCG architecture typically mirror the structure of the engineering workflows they support. Feedback loops and iterative refinement steps within such engineering workflows thus make DCGs the frontrunner architectural choice. We hereafter briefly detail how publications using a DCG MAS architecture facilitate alignment with the corresponding engineering workflow. Afterwards, we detail how previous work addresses tasks using a DCG that does not follow existing engineering workflows. Finally, we inspect how other architecture types, I, IVa, and IVb, are applied in previous work.

Vyas et al. [42] suggest mirroring the control and monitoring structure of processes at an industrial scale. This results in a DCG due to the nested feedback loops of common control structures. The DCG presented by Gowaikar et al. [43•] aims at the automatic creation of P&IDs in an iterative, cyclic workflow. The user describes individual subsystems of the envisioned P&ID in natural language, which the MAS then translates into a standardized, XML-based representation format for each. The representation of each P&ID subsystem is validated by a human-in-the-loop. This sequence of steps continues until all subsystems are added to the P&ID. Pajak et al. [45] propose a MAS for a decision-making process for operational setpoint selection of a gas-oil separation plant. The MAS architecture follows three steps: deriving a Pareto front of potential setpoints, selecting a suitable setpoint — considering both economic and environmental aspects, and finally, running a simulation of the plant with the selected setpoint. The decision-making process within the second step closely resembles an iterative human-centric workflow, thus resulting in a DCG MAS architecture: two agents enter a multi-turn debate, each supporting a different objective: minimal costs or minimal emissions. A third agent thereafter selects a setpoint based on the arguments presented during the debate. Srinivas et al. [48] employ a DCG architecture to systematically extract and aggregate chemical process information from a plethora of multimodal sources. A meta-agent orchestrates sub-agents that specialize in a certain modality each: images, journal articles, patents, HTML, etc. Each agent describes the information obtained from each modality in language. The sum of information is aggregated and assessed, introducing an iterative feedback loop.

Previous work also employs DCG architectures for workflows distinct from human-centric procedures [46•,47]. Srinivas et al. [46•] implement an iterative two-agent MAS that generates natural language descriptions of PFDs and P&IDs. A coordinating agent guides a subordinate agent to retrieve relevant data from curated

databases. The resulting response is evaluated iteratively by a critique agent that provides feedback using an LLM-as-a-judge mechanism or human feedback until a generated response satisfies the critique agent's judgment. Zeng et al. [47] define process operating constraints using a DCG that differs from heuristics-based human-centric workflows. Given a user's description of a process, an initial agent defines bounds for the process's decision variables. Next, a group of agents derives suitable parameter values in a guided, iterative collaborative workflow.

The chain (I) and DAG (II) architectures enforce strictly linear or branching workflows without feedback loops. For instance, Lee et al. [41•] use a six-agent chain (I) architecture following a predefined sequence of tasks to improve existing PFDs. Taskiran et al. [39•] employ two LLM-backed agents in sequence for entity extraction and response generation, respectively, as a chatbot for pharmaceutical engineering questions. However, the rigid design types (I-II) limit the system's ability to handle iterative refinement or adapt to edge cases, which may constrain performance on more complex or ill-defined tasks. In contrast, DCGs (III) offer a balance between rigidity, which enforces task structure, and flexibility, which allows iteration and adaptation to edge cases.

The centralized (IVa), the nested hierarchical (IVb), and the network architecture (V) favor flexibility over structured flow and align with tasks that lack a predefined engineering sequence. The centralized architecture (IVa) used by Sakhinana et al. [44] assigns decision-making to a central coordinator that delegates tasks to specialized agents. Zou et al. [49••] implement the nested hierarchical (IVb) architecture where a top-level agent delegates tasks to specialized subgroups of agents. This publication presents a hybrid form because elements of a network (V) are present: Some subgroup agents interact with agents at different hierarchy levels or agents of other subgroups.

Tool integration

A majority of MAS applications in chemical engineering rely on tool integration [41•–47,49••]. This highlights the central role of outsourcing tasks to specialized tools from purely LLM-based workflows. Tool integration is thus a crucial enabler in establishing MAS functionality within chemical engineering.

Previous work chose different ways to interface with the tools listed in Table 1, depending on the public availability of tool interfaces. Existing Application Programming Interfaces (APIs) are leveraged by previous work to integrate desired tools into chemical engineering MASs, for example, Stack Overflow's and Wolfram Alpha's APIs [44], or Microsoft Visio's C# API [43•]. If offline software

does not provide publicly available interfaces, users must develop custom interfaces to facilitate tool integration. For example, Pajak et al. [45] and Zeng et al. [47] interact with simulation environments through tailored interfaces to Aspen HYSYS and Pyomo [45], and IDAES [47], respectively.

The distribution of tools across agents is a critical decision that is addressed differently in the reviewed literature. In single-agent systems, all tools are assigned to a single agent, for example, in the case of ChemCrow [19••]. ChemCrow's LLM-powered agent can access 18 tools, requiring it to both select the appropriate tool and formulate the corresponding tool-call arguments to accomplish tasks spanning from automatic synthesis to materials design. In most reviewed MAS examples, tools are distributed across agents by expertise. For example, El Agente [49••] groups agents into modules by expertise. A subagent may have access to a selection of tools that match the domain of expertise, or no tool access. An individual tool may be called by different agents if the tool suits the agents' scope of action, for example, the execution of Python code in El Agente's architecture [49••].

Database integration

Across the reviewed literature, external information from databases is retrieved to enhance the domain relevance of LLM-backed agents' prompts. The works differ in how information is retrieved from the external databases. The most common retrieval methods are RAG [45] and Graph RAG [46•48]. Such RAG-based approaches aim at retrieving information from the external vector database that is semantically similar to a provided query. Pajak et al. [45] extract relevant pieces of text from larger reports, for example, the COP28. Graph RAG database retrieval is applied for advanced code generation [44] and retrieval of process flow and instrumentation descriptions [46•,48]. A different approach to database integration is presented by Taskiran et al. [39•], where raw information is queried from a graph database.

We observe differences in the generation of the databases used in the reviewed literature. The chemical engineering domain is not a classical big data domain [50] as there is a shortage of publicly available relational and graph databases compared to other domains, such as chemistry. Hence, previous work presents approaches to generate tailored structured databases. The source of origin for all databases used by the reviewed literature is unstructured literature [39•,44–46,48,51]. Most of these works [39•,44,46•,48,51] construct a knowledge graph of text embeddings of the information acquired. Before an extensive knowledge graph can be built, extensive amounts of multimodal information are aggregated first [44,46•,48,51]. The modalities include text in combination with equations [51], code [44], tables [44,51], and

figures [44,48,51]. To generate text embeddings of all modalities, previous literature commonly employs multimodal LLMs to produce descriptions of images, tables, etc. [44,48,51]. The ontology of the underlying knowledge graph is often extracted dynamically by an LLM [44,48]. An exception to this is the work of Taskiran et al. [39•], who base their knowledge graph construction process on unified, domain-specific ontologies.

Modalities

Chemical engineering spans heterogeneous data types: sensor signals, molecular representations (e.g. SMILES, graphs), process flowsheets, P&IDs, and simulation files [52]. Textual information (natural language instructions or additional contextual information) is often inserted directly into LLM prompts. Modalities other than text are typically preprocessed before they are introduced into the MAS or individual LLMs. In the previous section on database integration, we outlined that previous work aggregates information stored in different modalities in academic literature into a joint knowledge graph of text embeddings [44,46•,48,51]. Tables, figures, and code are summarized using multimodal LLMs [44,46•,48,51]. Numerical values — such as simulation results [45,47] or sensor data [42] — are typically embedded into prompts. Chemical processes are represented via natural language descriptions in the work of Srinivas et al. [46•]. Chemical processes can also be represented as graphs, which are used either as an alternative to text (e.g. D-SFILES [41•]) or as structured XML inputs [43•,53].

Human integration

Most MASs in chemical engineering reviewed so far operate autonomously, and they do not integrate human oversight [41•,42,44,45,47]. Four studies embed human interaction into the MAS architecture [43•,46•,48,49••]: Gowaikar et al. [43•] employ a human assessor to validate new additions to a P&ID visually through MS Visio. Srinivas et al. [46•] and Srinivas et al. [48] use a human-in-the-loop to provide feedback on suggestions made during the solution process of the MAS. Zou et al. [49••] present a versatile graphical user interface for human interaction and guidance.

Most works reviewed so far focus on solving a specific task in chemical engineering. A notable exception is the work of Sakhinana et al. [44], which addresses general question answering in chemical engineering. In the next section, we outline our perspective on developing a general MAS for chemical engineering.

Perspective

We envision MASs to become an important tool for chemical engineering (**Our vision**). After proposing our vision, we discuss integral parts to achieve this vision, including MAS architecture (**Architecture**), tool (**Tool**

integration), and database integration (**Database integration**), knowledge base integration (**Knowledge base integration**), and chemical engineering foundation models with domain-specific modalities (**Chemical engineering foundation models with domain-specific modalities**). In Section **Transparency, safety, and responsible development**, we critically reflect on challenges regarding the transparency and reliability of such systems.

Our vision

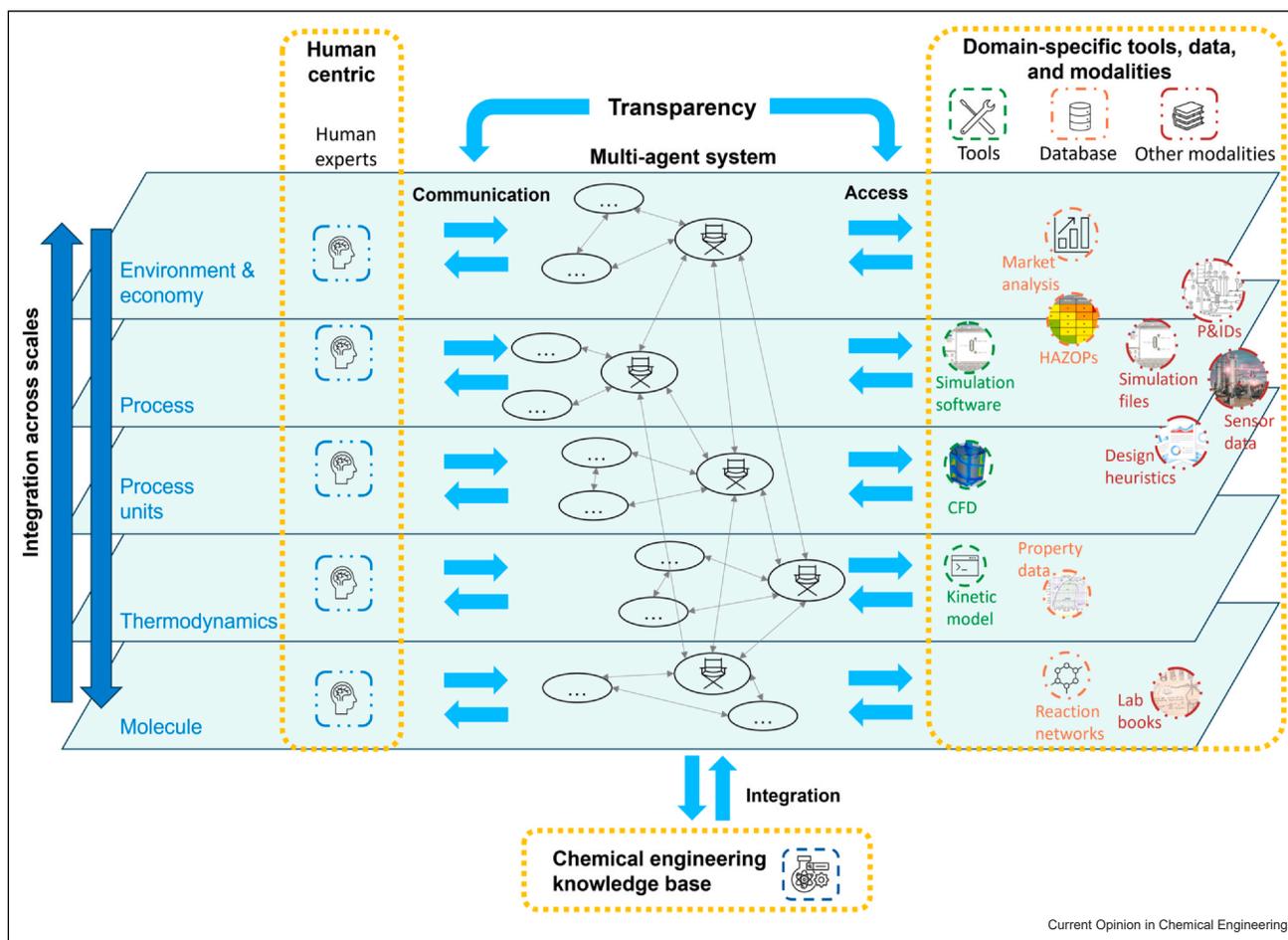
We envision MASs in chemical engineering as interconnected, human-centric collaborators that integrate across different scales, with access to domain-specific tools, databases, and modalities to drive intelligent and transparent decision-making.

Chemical engineering challenges span multiple scales from molecular to plant-wide operations and global supply chains. As illustrated in **Figure 2**, MASs offer a natural way to bridge these layers, with agents specializing in specific tasks while coordinating toward shared objectives. This architecture aligns closely with how chemical engineering work is already organized: in teams of experts using specialized tools and information. For example, agents equipped with thermodynamic and kinetic models can explore reaction pathways and materials, accelerating discovery at the molecular level. At the process level, operational agents dynamically run optimization routines, balancing yield, energy efficiency, and safety. Across plant and supply chain scales, agents negotiate schedules, logistics, and inventories in real time, while sustainability-focused agents continuously evaluate environmental and economic performance.

We envision a future where every engineer manages a team of intelligent agents. This paradigm extends existing workflows and empowers engineers in a scalable, intuitive way. At all times, human engineers remain central to this ecosystem. Rather than replacing human expertise, MASs are designed to support and augment it. This requires agents to communicate in ways that are transparent, interpretable, and aligned with the users, for example, through natural language, engineering diagrams, modeling code, and experimental procedures.

Realizing this vision depends on agents being well integrated into the chemical engineering domain. This includes interoperability with modeling and simulation tools, access to validated databases, and the ability to interpret diverse data modalities. The goal is not to build generic end-to-end black-box AI systems, but domain-specialized teams of agents that understand and respect the intricacies of their subdomain within chemical engineering. The following subsections outline the key building blocks and developments that are required to realize our vision.

Figure 2



MASs in chemical engineering as interconnected, human-centric collaborators that integrate across different scales, with access to domain-specific tools, databases, and modalities to drive intelligent and transparent decision-making.

Architecture

The architecture of a MAS is a key design decision as it substantially influences the conversation workflows, adaptability, and overall performance [12,15,54,55]. Previous research has shown that architectural choices are a major contributing factor to the success of MASs in other domains [26,54,56]. For example, Ishibashi et al. [26] proposes a flexible, hierarchical, dynamic, and scalable architecture for code generation and optimization, which yields competitive performance on a domain-specific benchmark. Chen et al. [54] explore four architecture types — centralized, decentralized, and two hybrid architectures — for the task of multi-robot coordination and find that a hybrid architecture yields best results and offers beneficial scaling opportunities for an increasing number of agents. Li et al. [56] propose AgentForest, a method that repeatedly samples responses from a single language model or a multi-agent framework, and then applies majority voting to select

the final answer. The authors compare the performance of AgentForest with common prompt-engineering techniques, multi-agent collaboration approaches, and hybrids that combine those methods with AgentForest [56]. Therefore, identifying suitable MAS architectures for chemical engineering is an important yet open research challenge.

Previous MAS architectures in chemical engineering have mirrored the existing engineering workflows by mapping human and agent roles (cf. [Multi-agent systems in chemical engineering](#)). Such MAS architectures promote transparency and interpretability while aiming to improve efficiency and enable (semi-)automation. MASs that resemble human-centric workflows are evocative of the concept of expert systems. Expert systems in chemical engineering have a long-standing tradition, dating back over five decades [50]. Expert systems described in literature pre-2000 strongly align with a vision of a

transparent and interpretable AI that is grounded in engineering knowledge, for instance, in the cases of physical property prediction [57] and catalyst selection [58]. Such expert systems contain valuable engineering knowledge that should be considered when building MASs. It can be argued that MASs can potentially receive soft engineering knowledge via their architecture if it follows expert systems' behaviors for certain tasks.

An additional approach to following existing engineering workflows could be a modular design that enables agents to coordinate flexibly while maintaining transparency and accountability. This modular design makes the MAS extendable to additional tasks without modifying existing components, unlike end-to-end workflows. In adjacent domains, like quantum chemistry [49••] and more distant domains, like computer science [56,59], there exist MAS architectures that successfully implement such a modular approach. Some works even use dynamic assembly of agents depending on the task [26]. We recommend exploring how MASs that innovate beyond established workflows compare to those that closely mirror human workflows. In particular, the hybridization between the two approaches could help tap the full potential of MASs in chemical engineering.

Tool integration

Tool integration is critical for MASs in chemical engineering, a domain reliant on specialized software across scales, that is, from DFT at the molecular level to process simulators like Aspen Plus and gPROMS. Agents must interface with these tools via standardized endpoints (e.g. Python APIs), yet most commercial software lacks such access, posing a key barrier [9]. Therefore, developing open interfaces and data standards is essential. We envision MASs with tool-aware agents: some wrap specific software, while others orchestrate tools for higher-level reasoning. Integration must be built into MAS architectures from the start to ensure scalability, interpretability, and reuse. Future work should examine agent-tool interaction limits, architectural patterns, and strategies for managing tool complexity transparently for both agents and users. Moreover, LLMs might require fine-tuning on our domain modeling languages to take full advantage of modeling environments beyond mere tool calls [60].

Database integration

Access to structured, validated data is essential for MASs in chemical engineering, supporting accurate decision-making, reducing hallucinations, and enabling transparent reasoning. However, early MAS efforts have not yet considered database integration, increasing reliance on internal memory or LLM inference, which is prone to errors and unverifiable outputs. We envision MASs interfacing with both open-access and commercial databases across scales: molecular (e.g. QM9, PubChem), reactions (e.g. Open

Reaction Database), thermodynamic (e.g. Design Institute for Physical Properties, Dortmund Data Bank), and kinetic (e.g. National Institute of Standards and Technology). Besides these databases, the engineering and operational data of plants should also be integrated, for example, through process historians and Asset Administration Shells. Future work should prioritize developing agents that access these databases via structured APIs or standardized interfaces. Embedding such capabilities early in MAS design will improve modularity, reasoning accuracy, and alignment with engineering practice.

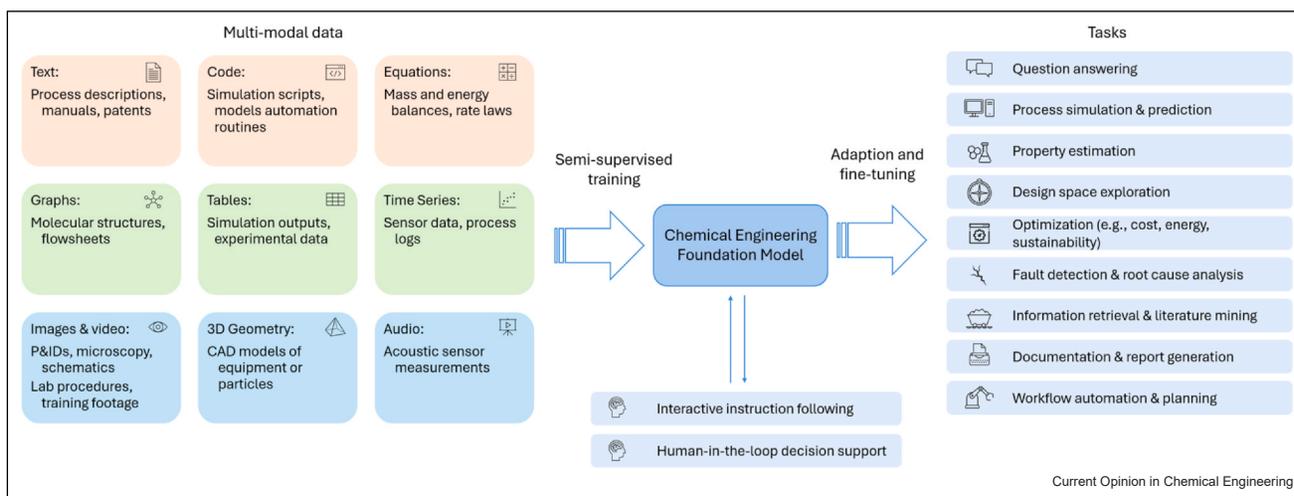
Knowledge base integration

Integrating chemical engineering knowledge into MASs is indispensable for reliable decision-making and coordination across tasks and tools. The failure of general-purpose LLMs at chemical engineering tasks can be attributed to multiple factors, such as inadequate contextual comprehension of domain-specific tokens and jargon, relationships, implicit attributes, and sensitivity to noise in their training data. An open challenge is how to represent and inject existing domain knowledge in a way that remains consistent, traceable, and extensible as systems evolve. A promising approach is to embed a knowledge graph within the MAS to provide a shared, machine-readable schema of concepts, entities, and relations. Domain ontologies such as OntoCape and DEXPI define standardized vocabularies and formal relationships that enable semantic interoperability, provenance tracking, and formal reasoning over domain knowledge. When implemented as knowledge graphs, these ontologies provide a machine-readable data structure that ensures consistency across applications, traceability of data and knowledge, and alignment with domain-specific constraints. The reviewed literature contains the first examples of such knowledge base integration within the chemical engineering domain [39•,44,46•,48,51]. However, only one work [39•] uses a fixed, domain-specific ontology.

Chemical engineering foundation models with domain-specific modalities

For MASs to effectively support chemical engineering workflows, the underlying AI must process the field's diverse data modalities (cf. [Multi-agent systems in chemical engineering](#)). AI cannot assist reliably if it cannot interpret the types of information engineers use daily. Previous approaches to multimodal integration made use of vision language models, for example, to extract figure descriptions [46•,51] or for the interpretation of scanning electron microscopy images [61]. However, general-purpose and out-of-the-box multimodal LLMs have so far not shown sufficient capability to reliably handle various domain-specific modalities in a chemical engineering context (cf. failure of visual language models in chemistry [62]). Current out-of-the-box LLMs are not sufficiently aware of the chemical engineering context and the domain-specific data modalities. A potential

Figure 3



Modalities that need to be integrated into a chemical engineering foundation model. On the left are common modalities and corresponding examples in the chemical engineering domain. A foundation model can be trained in a semi-supervised way. The foundation model can then be customized toward more specific downstream tasks.

approach could be to tailor domain foundation models via training or fine-tuning. Foundation models, in contrast to LLMs, are designed to perform a wider range of tasks on various modalities beyond text understanding and generation. Chemical engineering foundation models could be used as agents within MASs to help the integration of siloed systems by bridging modality gaps.

We propose developing foundation models trained on chemical engineering-specific multimodal data (cf. Figure 3). These models should incorporate multimodal learning, leveraging both labeled and unlabeled datasets through semi-supervised routines. Such semi-supervised learning approaches hold strong potential for fine-tuning in low-data regimes, which are common in chemical engineering [50]. In particular, we envision that graphs will play a key role as a data modality for chemical engineering [63,64], building on recent advances in computer science [65–69]. Ontology-backed knowledge graphs can further reduce hallucinations through formal constraints and rule-based validation. To get the engineering data in a machine-readable format, digitization efforts may be necessary [70]. For industry use, compact models (e.g. small language models) or federated learning approaches may be necessary.

Transparency, safety, and responsible development

MASs must meet the highest standards for transparency, reliability, and accountability, given the safety-critical nature of chemical processes. General-purpose LLMs, while powerful, are prone to hallucinations and lack grounding in physical laws, making them unsuitable as fully autonomous decision-makers in this domain. Specifically, the auto-complete output generation of

established encoder-decoder or decoder-only LLMs can cause the propagation of hallucinations in certain application scenarios. Rigorous evaluation of LLM-generated content is always required because outputs that appear coherent and plausible may be incorrect.

Rather than aiming for full automation, MASs should be designed as symbiotic human-AI collaborations. This includes static human-in-the-loop checkpoints and dynamic user consultation during inference — design principles that foster trust, ensure accountability, and align with emerging regulations like the EU AI Act, which mandates human oversight for high-risk AI applications. We consider static monitoring of LLM-generated outputs to be feasible to a certain extent, for example, type-checking of tool arguments or enforcing limits on repeated tool calls. The monitoring of LLM outputs that influence safety-critical decisions should always require human approval, for example, whenever a MAS interfaces with the real world. Even when MASs do not influence high-stakes operation decision-making, the propagation of hallucinations might still cause significant time loss when outputs must be searched for potential errors by human engineers. Previous studies have implemented human-in-the-loop mechanisms, including visual interfaces for user approval [43•] and workflows that integrate iterative human feedback [46•,48]. Human involvement can also be included in the training or fine-tuning of LLMs already, for example, via reinforcement learning with human feedback [71] or through direct preference optimization [72].

MAS architectures must be interpretable and evaluable. While recent open-source benchmark datasets provide a

starting point [73••,74], chemical engineering workflows require domain-specific performance criteria and failure taxonomies. Broader adoption will depend on standardized, task-specific evaluation strategies beyond isolated case studies [41•,45]. Interpretable MAS architectures and their design process can be supported through visualization and analysis tools such as AgentCoord [75] or LangChain's LangSmith service. Finally, the development and deployment of LLM-based MASs must consider their computational and environmental costs. Training and operating large models consume significant energy, and these impacts should be assessed using life cycle analysis frameworks to ensure AI workflows support the broader sustainability goals of chemical engineering.

Conclusions

MASs are a very recent development in chemical engineering, with only a handful of pioneering studies published to date. Yet, they hold great promise to transform the entire field by essentially providing every engineer with a capable team of specialized agents. We surveyed MASs in chemical engineering, highlighting key developments in architecture, tool, and database integration, addressing knowledge integration across modalities, and human-in-the-loop interaction. Despite early promise, the development is only starting, and scientific challenges remain, ranging from designing suitable architectures and integrating tools and data to developing foundation models with domain-specific modalities. In addition, issues of transparency, safety, human oversight, and environmental impact must be addressed. As a young but fast-moving field, MASs offer exciting opportunities to rethink workflows in chemical engineering.

Data Availability

No data were used for the research described in the article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT 4o in order to correct the grammar and polish the sentences. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Acknowledgement

This research is supported by Shell Global Solutions International B.V. and the Netherlands Organisation for Scientific Research (NWO), Netherlands under the Veni grant (ID 20196), for which we express sincere gratitude.

References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

- of special interest
- of outstanding interest

1. OpenAI, et al. (2023). **GPT-4 Technical Report**. [10.48550/ARXIV.2303.08774](https://arxiv.org/abs/2303.08774).
2. S. Minaee, et al., **Large language models: A survey**, arXiv preprint arXiv:2402.06196 (2024).
3. Schweidtmann AM: **Generative artificial intelligence in chemical engineering**, *Nature. Chem Eng* 2024, **1**:193, <https://doi.org/10.1038/s44286-024-00041-5>
4. Decardi-Nelson B, et al.: **Generative artificial intelligence in chemical engineering spans multiple scales**. *Front Chem Eng* 2024, **6**:1458156, <https://doi.org/10.3389/fceng.2024.1458156>
5. Decardi-Nelson B, et al.: **Generative AI and process systems engineering: the next frontier**. *Comput Chem Eng* 2024, **187**:108723, <https://doi.org/10.1016/j.compchemeng.2024.108723>
6. Woo T, et al.: **Leveraging generative ai and large language model for process systems engineering: A state-of-the-art review**. *Korean J Chem Eng* 2025, **42**:2787-2808, <https://doi.org/10.1007/s11814-025-00524-y>
7. Daoutidis P, et al.: **Machine learning in process systems engineering: challenges and opportunities**. *Comput Chem Eng* 2024, **181**:108523, <https://doi.org/10.1016/j.compchemeng.2023.108523>
8. Venkatasubramanian V, et al.: **Quo vadis chatgpt? from large language models to large knowledge models**. *Comput Chem Eng* 2025, **192**:108895, <https://doi.org/10.1016/j.compchemeng.2024.108895>.
This is a cornerstone perspective paper on the integration of domain-specific and physics knowledge into LLMs.
9. Du W, et al.: **The potential and challenges of large language model agent systems in chemical process simulation: from automated modeling to intelligent design**. *Front Chem Sci Eng* 2025, **19**:99, <https://doi.org/10.1007/s11705-025-2587-5>
10. S.J. Russell, Artificial intelligence, third edition, global edition Edition, Always learning, Pearson, Boston, 2016, "Authorized adaptation from the United States edition, entitled 'Artificial intelligence: a modern approach', third edition, ISBN 9780136042594, by Stuart J. Russell and Peter Norvig published by Pearson Education ©2010." - Rückseite der Titelseite. Boston: Pearson, 2016.11132.
11. Wooldridge M: **An introduction to multiagent systems**. John Wiley & Sons; 2009.
12. Liu B., et al., Advances and Challenges in foundation agents: From Brain-inspired Intelligence to Evolutionary, Collaborative, and Safe Systems, arXiv:2504.01990 (2025), <https://doi.org/10.48550/arXiv.2504.01990>
13. Wu B., et al., Autogen: Enabling Next-gen LLM Applications via
•• Multi-agent Conversation, arXiv:2308.08155 (2023), <https://doi.org/10.48550/arXiv.2308.08155>.
AutoGen makes two core contributions: (1) An argumentation baseline for the benefits of MAS in contrast to single-agent systems for general use-cases. (2) An open-source framework for customized implementations of MASs.
14. Liang T, et al.: **Encouraging divergent thinking in large language models through multi-agent debate**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Edited by Al-Onaizan Y, et al. Association for Computational Linguistics, Miami, Florida, USA; 2024:17889-17904, <https://doi.org/10.18653/v1/2024.emnlp-main.992>

15. Tran K.T., et al., Multi-Agent Collaboration Mechanisms: A Survey of LLMs, arXiv:2501.06322 (2025), <https://doi.org/10.48550/arXiv.2501.06322>
16. He J, et al.: Llm-based multi-agent systems for software engineering: literature review, vision, and the road ahead. *ACM Trans Softw Eng Methodol* 2025, **34**:1-30, <https://doi.org/10.1145/3712003>
17. Ghafarollahi A, et al.: Protagents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digit Discov* 2024, **3**:1389-1409, <https://doi.org/10.1039/d4dd00013g>
18. Boiko DA, et al.: Autonomous chemical research with large language models. *Nature* 2023, **624**:570-578, <https://doi.org/10.1038/s41586-023-06792-0>
19. Bran AM, et al.: Augmenting large language models with chemistry tools. *Nat Mach Intell* 2024, **6**:525-535, <https://doi.org/10.1038/s42256-024-00832-8> (ChemCrow is one of the first groundbreaking works showcasing the applicability of LLMs in the chemistry domain, outsourcing critical tasks to a multitude of tools). ChemCrow is one of the first groundbreaking works showcasing the applicability of LLMs in the chemistry domain, outsourcing critical tasks to a multitude of tools.
20. Y. Fehlis, et al. Accelerating Drug Discovery Through Agentic AI: A Multi-Agent Approach to Laboratory Automation in the DMTA cycle; 2025, [10.48550/ARXIV.2507.09023](https://doi.org/10.48550/ARXIV.2507.09023).
21. J. Tian, et al. A Multi-agent Framework Integrating Large Language Models and Generative AI for Accelerated Metamaterial Design; 2025, [10.48550/ARXIV.2503.19889](https://doi.org/10.48550/ARXIV.2503.19889).
22. J. Gottweis, et al., Towards an AI Co-scientist; 2025, [10.48550/ARXIV.2502.18864](https://doi.org/10.48550/ARXIV.2502.18864).
23. C. Lu, et al., The AI Scientist: Towards Fully Automated Open-ended Scientific Discovery; 2024, [10.48550/ARXIV.2408.06292](https://doi.org/10.48550/ARXIV.2408.06292).
24. Y. Yamada, et al. The Ai Scientist-v2: Workshop-level Automated Scientific Discovery via Agentic Tree Search; 2025, [10.48550/ARXIV.2504.08066](https://doi.org/10.48550/ARXIV.2504.08066).
25. X. Tang, et al., Medagents: Large Language Models as Collaborators for Zero-Shot Medical Reasoning; 2023, [10.48550/ARXIV.2311.10537](https://doi.org/10.48550/ARXIV.2311.10537).
26. Y. Ishibashi, et al., Self-organized Agents: A LLM Multi-agent Framework Toward Ultra Large-scale Code Generation and Optimization; 2024, [10.48550/ARXIV.2404.02183](https://doi.org/10.48550/ARXIV.2404.02183).
27. Boskabadi MR, et al.: Industrial agentic ai and generative modeling in complex systems. *Curr Opin Chem Eng* 2025, **48**:101150, <https://doi.org/10.1016/j.coche.2025.101150>
28. Rothfarb S, et al.: Multi-agent large language model frameworks: unlocking new possibilities for optimizing wastewater treatment operation. *Environ Res* 2025, **275**:121401, <https://doi.org/10.1016/j.envres.2025.121401>
29. Z. Hong, et al., Next-generation Database Interfaces: A Survey of LLM-based text-to-SQL; 2024, [10.48550/ARXIV.2406.08426](https://doi.org/10.48550/ARXIV.2406.08426).
30. M.G. Ozsoy, et al., Text2cypher: Bridging Natural Language and Graph Databases; 2024, [10.48550/ARXIV.2412.10064](https://doi.org/10.48550/ARXIV.2412.10064).
31. P. Lewis, et al., Retrieval-augmented Generation for Knowledge-intensive NLP Tasks; 2020, [10.48550/ARXIV.2005.11401](https://doi.org/10.48550/ARXIV.2005.11401).
32. D. Edge, et al., From Local to Global: A Graph Rag Approach to Query-Focused Summarization; 2024, [10.48550/ARXIV.2404.16130](https://doi.org/10.48550/ARXIV.2404.16130).
33. H. Han, et al., Retrieval-augmented Generation with Graphs (graphrag); 2025, [10.48550/ARXIV.2501.00309](https://doi.org/10.48550/ARXIV.2501.00309).
34. Y. Hu, et al., Grag: Graph Retrieval-augmented Generation; 2024, [10.48550/ARXIV.2405.16506](https://doi.org/10.48550/ARXIV.2405.16506).
35. Inc. CrewAI™ (2025). (<https://www.crewai.com/>).
36. LangChain Inc. (2025). (<https://github.com/langchain-ai/langgraph>).
37. G. Li, et al., CAMEL: Communicative Agents for “mind” exploration of Large Language Model Society; 2023, [10.48550/ARXIV.2303.17760](https://doi.org/10.48550/ARXIV.2303.17760).
38. S. Hong, et al., MetaGPT: Meta Programming for a Multi-agent Collaborative Framework; 2023, [10.48550/ARXIV.2308.00352](https://doi.org/10.48550/ARXIV.2308.00352).
39. Taskiran NP, et al.: A knowledge-graph-based pharmaceutical engineering chatbot for drug discovery. *Comput Chem Eng* 2025, **203**:109318, <https://doi.org/10.1016/j.compchemeng.2025.109318> (This work presents an approach to integrate domain knowledge into an LLM-based chatbot by interacting with a knowledge graph that is based on established, domain-specific ontologies). This work presents an approach to integrate domain knowledge into an LLM-based chatbot by interacting with a knowledge graph that is based on established, domain-specific ontologies.
40. S. Han, et al., LLM Multi-agent Systems: Challenges and Open Problems; 2024, [10.48550/ARXIV.2402.03578](https://doi.org/10.48550/ARXIV.2402.03578).
41. Lee D, et al.: Gpt prompt engineering for a large language model-based process improvement generation system. *Korean J Chem Eng* 2024, **41**:3263-3286, <https://doi.org/10.1007/s11814-024-00276-1> (This work is one of the earliest applications of a multi-agent system in the chemical engineering domain showcasing the subdivision of tasks among different agents). This work is one of the earliest applications of a multi-agent system in the chemical engineering domain showcasing the subdivision of tasks among different agents.
42. J. Vyas, et al., Autonomous Industrial Control Using an Agentic Framework with Large Language Models; 2024, [10.48550/ARXIV.2411.05904](https://doi.org/10.48550/ARXIV.2411.05904).
43. S. Gowaikar, et al., An Agentic Approach to Automatic Creation of P&ID Diagrams from Natural Language Descriptions; 2024, [10.48550/ARXIV.2412.12898](https://doi.org/10.48550/ARXIV.2412.12898). The authors devise a flexible approach to P&ID generation. In addition, this work includes a visual human-in-the-loop validation step.
44. Sakhinana SS, et al.: Knowledge graph modeling-driven large language model operating system (llm os) for task automation in process engineering problem-solving. *Proc AAI Symp Ser* (1) 2024, **4**:222-232, <https://doi.org/10.1609/aaais.v4i1.31796>
45. Pajak E, et al.: Multi-agent llms for automating sustainable operational decision-making. Proceedings of the 35th European Symposium on Computer Aided Process Engineering (ESCAPE 35), Vol. 4 of ESCAPE 35. PSE Press.; 2025:1824-1829, <https://doi.org/10.69997/sct.156776>
46. S.S. Srinivas, et al., Autochemschematic ai: A closed-loop, physics-aware agentic framework for auto-generating chemical process and instrumentation diagrams (2025). [10.48550/ARXIV.2505.24584](https://doi.org/10.48550/ARXIV.2505.24584). This work makes three core contributions: (1) The generation of a graph database of process descriptions for the use in a graph RAG approach. (2) The authors fine-tune small language models (SLMs) for specialized downstream tasks. (3) Simulator-based validation of content generated by a MAS.
47. T. Zeng, et al., Llm-guided chemical process optimization with a multi-agent approach (2025). [10.48550/ARXIV.2506.20921](https://doi.org/10.48550/ARXIV.2506.20921).
48. S.S. Srinivas, et al., Accelerating manufacturing scale-up from material discovery using agentic web navigation and retrieval-augmented ai for process engineering schematics design (2024). [10.48550/ARXIV.2412.05937](https://doi.org/10.48550/ARXIV.2412.05937).
49. Zou Y, et al.: El agente: An autonomous agent for quantum chemistry. *Matter* (7) 2025, **8**:102263, <https://doi.org/10.1016/j.matt.2025.102263> (The authors present a highly relevant example of a performative, large-scale multi-agent architecture that facilitates the integration of a multitude of specialized tools to address a wide range of domain-specific tasks). The authors present a highly relevant example of a performative, large-scale multi-agent architecture that facilitates the integration of a multitude of specialized tools to address a wide range of domain-specific tasks.
50. Venkatasubramanian V: The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE J* (2) 2018, **65**:466-478, <https://doi.org/10.1002/aic.16489>

51. Ovalle D, et al.: **Grapse: Graph-based retrieval augmentation for process systems engineering**. Proceedings of the 35th European Symposium on Computer Aided Process Engineering (ESCAPE 35), Vol. 4 of ESCAPE 35. PSE Press,; 2025:1598-1604, <https://doi.org/10.69997/sct.198790>
52. Schweidtmann AM, et al.: **Machine learning in chemical engineering: A perspective**. *Chem Ing Tech* (12) 2021, **93**:2029-2039.
53. Alimin AA, et al.: **Talking like piping and instrumentation diagrams (p&ids)**. Proceedings of the 35th European Symposium on Computer Aided Process Engineering (ESCAPE 35). Vol. 4. ESCAPE 35. PSE Press,; 2025:1676-1681, <https://doi.org/10.48550/ARXIV.2502.18928>
54. Y. Chen, et al., **Scalable multi-robot collaboration with large language models: Centralized or decentralized systems?**, *The 2024 International Conference on Robotics and Automation* (2023). [10.48550/ARXIV.2309.15943](https://doi.org/10.48550/ARXIV.2309.15943).
55. J. Fang, et al., **A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems** (2025). [10.48550/ARXIV.2508.07407](https://doi.org/10.48550/ARXIV.2508.07407).
56. J. Li, et al., **More agents is all you need** (2024). [10.48550/ARXIV.2402.05120](https://doi.org/10.48550/ARXIV.2402.05120).
57. Bañares-Alcántara R, Westerberg AW, Rychener MD: **Development of an expert system for physical property predictions**. *Comput Chem Eng* (2) 1985, **9**:127-142, [https://doi.org/10.1016/0098-1354\(85\)85003-1](https://doi.org/10.1016/0098-1354(85)85003-1)
58. Bañares-Alcántara R, et al.: **Decade— a hybrid expert system for catalyst selection—II. final architecture and results**. *Comput Chem Eng* (9-10) 1988, **12**:923-938, [https://doi.org/10.1016/0098-1354\(88\)87018-2](https://doi.org/10.1016/0098-1354(88)87018-2)
59. W. Chen, et al., **Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors** (2023). [10.48550/ARXIV.2308.10848](https://doi.org/10.48550/ARXIV.2308.10848).
60. S. Rupprecht, et al., **Text2model: Generating dynamic chemical reactor models using large language models (llms)**, arXiv preprint arXiv:2503.17004 (2025).
61. A.S. Alshehri, et al., **Multimodal deep learning for scientific imaging interpretation** (2023). [10.48550/ARXIV.2309.12460](https://doi.org/10.48550/ARXIV.2309.12460).
62. N. Alampara, et al., **Probing the limitations of multimodal language models for chemistry and materials research** (2024). [10.48550/ARXIV.2411.16955](https://doi.org/10.48550/ARXIV.2411.16955).
63. Gao Q, et al.: **Deep reinforcement learning for process design: Review and perspective**. *Curr Opin Chem Eng* 2024, **44**:101012, <https://doi.org/10.1016/j.coche.2024.101012>
64. Balhorn LS, et al.: **Graph-to-files: Control structure prediction from process topologies using generative artificial intelligence**. *Comput Chem Eng* 2025,109121.
65. S. Ma, et al., **Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation** (2024). [10.48550/ARXIV.2407.10805](https://doi.org/10.48550/ARXIV.2407.10805).
66. B. Perozzi, B. Fatemi, D. Zelle, A. Tsitsulin, M. Kazemi, R. Al-Rfou, J. Halcrow, **Let your graph do the talking: Encoding structured data for llms** (2024). [10.48550/ARXIV.2402.05862](https://doi.org/10.48550/ARXIV.2402.05862).
67. B. Jin, et al., **Large language models on graphs: A comprehensive survey**, *Transactions on Knowledge and Data Engineering (TKDE) 2024* (2023). [10.48550/ARXIV.2312.02783](https://doi.org/10.48550/ARXIV.2312.02783).
68. Pan S, et al.: **Unifying large language models and knowledge graphs: A roadmap**. *IEEE Trans Knowl Data Eng (TKDE) 2024* (7) 2023, **36**:3580-3599, <https://doi.org/10.1109/tkde.2024.3352100>
69. L. Luo, et al., **Reasoning on graphs: Faithful and interpretable large language model reasoning** (2023). [10.48550/ARXIV.2310.01061](https://doi.org/10.48550/ARXIV.2310.01061).
70. Theisen MF, et al.: **Digitization of chemical process flow diagrams using deep convolutional neural networks**. *Digit Chem Eng* 2023, **6**:100072, <https://doi.org/10.1016/j.dche.2022.100072>
71. P. Christiano, J. Leike, T.B. Brown, M. Martic, S. Legg, D. Amodei, **Deep reinforcement learning from human preferences** (2017). [10.48550/ARXIV.1706.03741](https://doi.org/10.48550/ARXIV.1706.03741).
72. R. Rafailov, et al., **Direct preference optimization: Your language model is secretly a reward model** (2023). [10.48550/ARXIV.2305.18290](https://doi.org/10.48550/ARXIV.2305.18290).
73. M. Cemri, et al., **Why do multi-agent llm systems fail?** (2025). [10.48550/ARXIV.2503.13657](https://doi.org/10.48550/ARXIV.2503.13657).
- This work categorizes failure modes of MAS on the basis of a comprehensive taxonomy.
74. F. Grötschla, et al., **Agentsnet: Coordination and collaborative reasoning in multi-agent llms** (2025). [10.48550/ARXIV.2507.08616](https://doi.org/10.48550/ARXIV.2507.08616).
75. B. Pan, et al., **Agentcoord: Visually exploring coordination strategy for llm-based multi-agent collaboration** (2024). [10.48550/ARXIV.2404.11943](https://doi.org/10.48550/ARXIV.2404.11943).