

From Text to Simulation: A Multi-Agent LLM Workflow for Automated Chemical Process Design

Xufei Tian¹, Wenli Du^{1,2*}, Shaoyi Yang¹, Han Hu¹, Hui Xin¹, Shifeng Qu¹, Ke Ye¹

¹State Key Laboratory of Industrial Control Technology, East China University of Science and Technology

²Huzhou Institute of Industrial Control Technology, Huzhou, China

{xufei.tian, sy.yang, han.hu, xinhui, shifengqu, yeke}@mail.ecust.edu.cn,

wldu@ecust.edu.cn

Abstract

Process simulation is a critical cornerstone of chemical engineering design. Current automated chemical design methodologies focus mainly on various representations of process flow diagrams. However, transforming these diagrams into executable simulation flowsheets remains a time-consuming and labor-intensive endeavor, requiring extensive manual parameter configuration within simulation software. In this work, we propose a novel multi-agent workflow that leverages the semantic understanding capabilities of large language models (LLMs) and enables iterative interactions with chemical process simulation software, achieving end-to-end automated simulation from textual process specifications to computationally validated software configurations for design enhancement. Our approach integrates four specialized agents responsible for task understanding, topology generation, parameter configuration, and evaluation analysis, respectively, coupled with Enhanced Monte Carlo Tree Search to accurately interpret semantics and robustly generate configurations. Evaluated on Simona, a large-scale process description dataset, our method achieves a 31.1% improvement in the simulation convergence rate compared to state-of-the-art baselines and reduces the design time by 89.0% compared to the expert manual design. This work demonstrates the potential of AI-assisted chemical process design, which bridges the gap between conceptual design and practical implementation. Our workflow is applicable to diverse process-oriented industries, including pharmaceuticals, petrochemicals, food processing, and manufacturing, offering a generalizable solution for automated process design.

Introduction

The design of chemical processes forms the foundation of numerous industrial sectors, including petrochemicals, pharmaceuticals, sustainable energy, and material manufacturing. Despite decades of development in process simulation software, translating high-level process requirements into executable simulation models remains a significant challenge in chemical engineering workflows (Mann, Gani, and Venkatasubramanian 2023). Engineers typically spend weeks transforming conceptual designs into detailed configurations, manually specifying hundreds of interdependent

parameters. This manual configuration process not only consumes valuable engineering time, but also limits exploration of innovative design alternatives.

Current approaches to chemical process design automation have achieved progress in specific aspects, yet they fail to provide comprehensive solutions. As illustrated in the middle section of Figure 1, existing methods typically focus on intermediate representations, either generating process flow diagrams or constructing graph-based representations (Balhorn et al. 2022; d’Anterrosches 2006). These approaches terminate at the structural representation level, leaving a critical gap between abstract process descriptions and executable simulations. While flowsheet-based methods capture the visual topology of processes, and hypergraph representations encode connectivity relationships, neither can directly generate simulation-ready configurations executable in industrial software. This limitation forces engineers to manually bridge the gap between these representations and functional process models, perpetuating bottlenecks in design workflows. The fundamental challenge extends beyond simply representing the process structure to generating complete executable configurations that satisfy complex engineering constraints. Process simulation requires a precise specification of thermodynamic models, operating conditions, equipment parameters, and control strategies, details that cannot be adequately captured by structural representations alone. Moreover, the interdependencies among these parameters create a complex optimization space where changes in one unit operation cascade throughout the entire process. Existing methods focus on either topology generation or parameter optimization in isolation, failing to address this holistic challenge, resulting in designs that appear structurally sound but prove infeasible or suboptimal when implemented in simulation software.

To address these limitations, we propose a novel approach, as shown in Figure 1, which directly transforms natural language process descriptions into executable computational processes for simulation software. Unlike existing methods that terminate at intermediate representations, we achieve an end-to-end workflow from textual specifications to validated simulation configurations. Our work leverages large language models (LLMs) within a multi-agent workflow to interpret process requirements, construct appropriate topologies, configure detailed parameters, and validate de-

*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

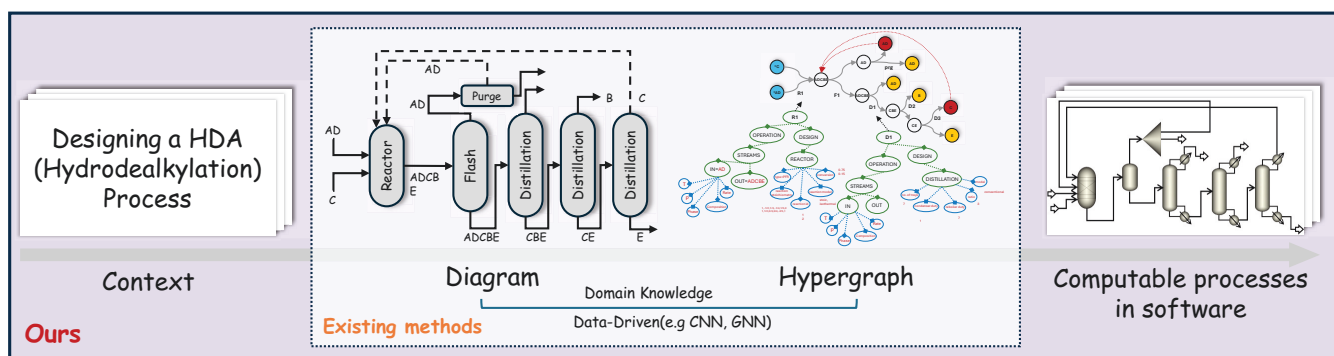


Figure 1: Representation of different stages of chemical process design workflow. “Diagram” refers to a general process flow diagram(Douglas 1988), while “Hypergraph” represents a process hypergraph with parameter annotations (Mann et al. 2024). The entire stage requires a lot of manual operation design, while existing automated methods only focus on various representations of graphs. Our method achieves cross-dimensional and fully automated implementation

signs through direct integration with professional simulation software. This approach eliminates the manual transformation steps required by current methods, enabling rapid iteration and exploration of design alternatives.

The key innovation of our approach lies in bridging the representation-to-implementation gap through three integrated components. First, we employ a multi-agent system that decomposes complex design tasks into specialized sub-tasks: task understanding, topology generation, parameter configuration, and evaluation analysis. Each agent leverages LLM capabilities while incorporating domain-specific constraints and engineering principles. Second, we introduce an Enhanced Monte Carlo Tree Search (E-MCTS) algorithm that efficiently explores the design space while maintaining feasibility constraints throughout the search process. Third, and most critically, we establish seamless bidirectional communication with industrial simulation software, enabling real-time validation and iterative refinement of generated configurations. This integration ensures that our outputs are not merely theoretical constructs but practically executable designs validated through rigorous computational verification. In summary, our contributions are as follows.

- We propose the first end-to-end workflow that directly generates executable chemical process simulations from natural language descriptions, bypassing the limitations of intermediate representations employed by existing methods.
- We develop a multi-agent architecture that effectively combines LLMs’ semantic understanding capabilities with chemical engineering domain knowledge, achieving accurate interpretation and implementation of complex process requirements.
- Through extensive experiments, we demonstrate that our work achieves an 80.3% success rate in generating valid executable process configurations, reducing design time by 89.0% compared to manual methods while maintaining industrial grade quality.
- We establish a new paradigm for AI-assisted chemical engineering that transcends structural representations

to deliver complete, simulatable solutions, opening new possibilities for accelerating process design innovation.

Related Work

Chemical Process Simulation

The traditional paradigm of chemical process simulation follows a multistage workflow: beginning with thermodynamic feasibility analysis, followed by process flow diagram (PFD) construction within feasible domains, and culminating in validation through specialized simulation software until convergence(Mann et al. 2022; Venkatasubramanian 2019; Tula et al. 2017). To accelerate this design cycle, researchers have explored digitization and symbolic representation approaches for flowsheets(Weininger 1988; Zhang, Sahinidis, and Siirola 2019; Tula, Eden, and Gani 2019). Early work employed traditional machine learning methods for automatic flowsheet completion(Vogel et al. 2023; Theisen et al. 2023), while eFILES(Mann et al. 2024) integrated domain knowledge with data-driven approaches to enable flowsheet analysis and weighting. Recent studies have incorporated large language models to enhance the intelligence of process synthesis(Vogel, Balhorn, and Schweidtmann 2023). The fundamental limitation of PFD representations lies in their level of abstraction: while they effectively capture unit operations and connectivity relationships, they omit critical configuration parameters essential for simulation convergence. Traditionally, setting parameters is a process that is not only time-consuming but also highly dependent on expert knowledge.

Existing automation approaches face three core challenges when addressing complex chemical processes: (i) lack of deep integration with specialized process simulation software, preventing rapid validation of generated design solutions; (ii) absence of simulation-based feedback mechanisms, precluding iterative optimization of configuration parameters; (iii) difficulty in bridging the semantic gap between high-level process descriptions and executable simulation configurations. This leaves numerous process designs at the conceptual level, with validation of large-scale pro-

cesses proving particularly challenging.

Agentic Workflow

The remarkable perception and reasoning capabilities of LLM-driven agents in complex environments have spurred extensive exploration and rapid advancement in multi-agent workflows (Song et al. 2023; Zhao et al. 2024; Xu et al. 2023; Liu et al. 2023; Ramesh et al. 2021; Yao et al. 2023; Wang et al. 2022). Current multi-agent workflow research spans both general-purpose and domain-specific applications (Pang et al. 2025; Zhuge et al. 2024b; Chen et al. 2025; Liu et al. 2024b,a), including automated workflow generation (Wu et al. 2021; Qiao et al. 2024; Li et al. 2024), code generation (Cen et al. 2025; Qian et al. 2023; Xu et al. 2023) and the collaboration of multi-agent (Zhang et al. 2024a; Chen et al. 2023; Li et al. 2023; Zhuge et al. 2024a). While ChemCrow (Bran et al. 2023) and Coscientist (Boiko et al. 2023) demonstrate the potential of LLMs in chemistry, a substantial gap persists between these approaches and industrial-grade applications: (i) insufficient consideration of engineering constraints and scale-up effects; (ii) inability to establish effective feedback loops with specialized engineering software. This gap motivates our focus on applying multi-agent workflows to the more challenging domain of industrial engineering implementation. Our approach processes textual descriptions to iteratively generate computationally verifiable chemical process configurations or thermodynamic analysis requests via multi-agent workflows, presenting significant challenges in configuration evaluation at each iteration.

Method

Overview

We propose a multi-agent collaborative workflow built upon LangGraph for automating chemical process design from natural language descriptions. The workflow transforms textual process specifications into executable simulation configurations through coordinated operation of specialized agents, employing E-MCTS for systematic exploration of the design space. Our system accepts natural language inputs ranging from high-level process requirements (e.g., “design an ethylene cracking process”) to detailed specifications encompassing equipment constraints, operating parameters, and economic or environmental objectives. It generates complete configuration files directly executable in professional chemical process simulation software, containing fully specified process topologies, operating conditions, and convergence results. As illustrated in Figure 2, our workflow coordinates four core agents: (1) *Task Understanding Agent*, (2) *Topology Generation Agent*, (3) *Parameter Configuration Agent*, (4) *Evaluation Analysis Agent*.

Our work consists of a main workflow accompanied by a subordinate workflow: the aforementioned main design workflow and an auxiliary thermodynamic analysis workflow executed prior to process design. This preliminary analysis validates critical component properties and phase equilibrium behavior, establishing a robust foundation for the main workflow and preventing fundamental errors in

subsequent iterations. The iterative refinement mechanism employs Enhanced Monte Carlo Tree Search to guide exploration, systematically navigating the design space while maintaining computational efficiency to ensure high-quality solutions within reasonable timeframes. This closed-loop optimization emulates human expert iterative refinement while leveraging computational advantages of automated exploration.

Task Understanding Agent

The Task Understanding Agent serves as the entry of our multi-agent workflow, designed to parse ambiguous user descriptions into structured requirements for downstream processing. Users typically provide natural language descriptions of their process design ideas, which often lack critical specifications necessary for chemical process configuration. For instance, when a user queries “Please help me check whether two substances form an azeotrope”, they are essentially requesting thermodynamic temperature properties of a binary system, though this technical requirement remains implicit in their description.

The fundamental challenge addressed by this agent is the semantic gap between informal user expressions and the rigorous specifications required by process simulation tools. To bridge this gap, we have developed a comprehensive data structure that encodes critical attribute information aligned with our simulation requirements. The extracted information is then structured into a standardized format that serves as the communication protocol between agents. This structured representation ensures that subsequent agents receive unambiguous specifications while preserving the flexibility to handle diverse user queries. Thus, the task understanding agent enables robust downstream processing and reduces the likelihood of configuration errors in the final simulation setup.

Topology Generation Agent

The Topology Generation Agent transforms process description into chemical process topologies, bridging the gap between abstract user requirements and executable flow-sheets. This agent achieves automated conversion from functional requirements to structural design by leveraging domain knowledge and systematic design principles, and is the only agent that does not participate in the thermodynamic analysis configuration workflow. The core design philosophy of this agent is to decouple topology generation from parameter configuration. Considering that the final output of the multi-agent workflow is a JSON configuration file verifiable by process simulation software, and that LLM-based agents struggle to directly generate configurations that strictly meet computational requirements, we have designed a specialized tool function library. This enables the agent to focus on high-level decisions regarding unit selection and connectivity relationships, thereby significantly improving design efficiency. The agent employs a multi-candidate strategy to generate several feasible alternatives, a feature that manifests in the subsequent E-MCTS. Each topology is represented as a directed graph $G = (V, E)$, where the node

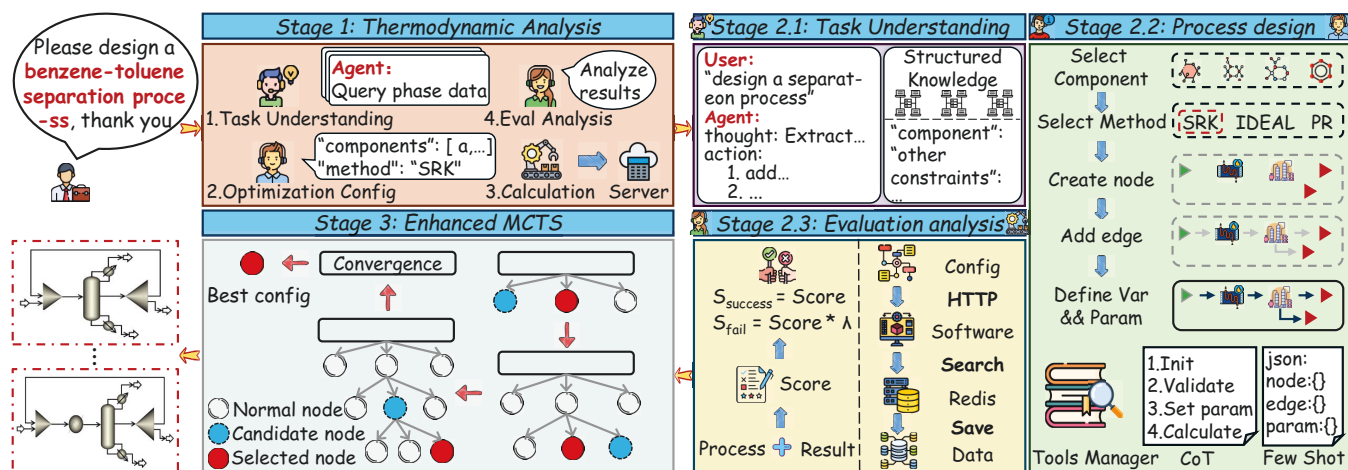


Figure 2: The overview of our comprehensive workflow, consisting of four key agents: (i) the *Task Understanding Agent*, which parses user inputs to extract key parameters, constraints, and design objectives. (ii) the *Topology Generation Agent*, which determines appropriate chemical components and thermodynamic methods, selects suitable unit operations and establishes their connections. (iii) the *Parameter Configuration Agent*, which assigns operating conditions to each unit within the generated topology, such as temperature and pressure. (iv) the *Evaluation Analysis Agent*, which assesses simulation results to determine whether to iterate or terminate.

set V corresponds to unit operations and the edge set E represents material flow connections. Each design is accompanied by a unique process manager reference identifier, enabling downstream agents to access and modify the topology structure to support the iterative optimization process.

The Topology Generation Agent accesses process design tools through a globally unified WorkflowToolsManager, implementing the following core functions: (i) Process initialization, creating independent process manager instances for each candidate solution to ensure parallel exploration of the design space; (ii) Adding components and selecting property methods; (iii) Unit operation instantiation, incorporating various equipment such as reactors, distillation columns, and heat exchangers; (iv) Stream connections, establishing material and energy flow relationships between unit operations; (v) Design persistence, saving completed topologies to a tree structure for subsequent processing. During subsequent iterations, this agent also supports cascading deletion of unit nodes and edge removal operations to facilitate topology design modifications.

Parameter Configuration Agent

The Parameter Configuration Agent leverages the powerful contextual understanding and reasoning capabilities of LLMs, integrating chemical engineering domain knowledge to transform topological structures into executable chemical processes by determining optimal operating parameters for each unit operation. This agent accomplishes two key functions: (i) Assigning initial operating conditions guided by a model template library; (ii) Performing further parameter adjustments based on simulation feedback. To enhance the rationality of parameter configurations, we employ chain-of-thought(Wei et al. 2022) prompting strategies to guide LLMs through step-by-step reasoning. Addi-

tionally, we have compiled extensive high-quality parameter configuration examples from feasible processes, utilizing few-shot learning to improve the quality of LLM-generated parameter configurations.

The agent passes adjusted configurations to dedicated computational nodes, interfacing with external simulation tools through asynchronous HTTP protocols, enabling rigorous validation via chemical process simulation software while maintaining a closed-loop optimization mechanism. This integration is crucial, as LLM-based agents, while excelling at high-level reasoning, struggle to directly generate parameters that strictly satisfy the complex interdependencies of chemical processes.

Evaluation Analysis Agent

The Evaluation Analysis agent serves as a critical quality assurance component in our workflow, fulfilling dual responsibilities: assessing current process configurations and providing actionable improvement directions for subsequent iterations. Following validation through external simulation software via computational nodes, the agent performs quantitative scoring of both the process configuration and simulation results. When simulations fail to converge, a penalty factor $\lambda \in (0, 1)$ is applied. The agent employs a comprehensive multi-dimensional evaluation framework that assesses process designs across the following key dimensions:

Economic Feasibility (\mathcal{E}_f) : This dimension evaluates the economic feasibility and cost-effectiveness of process configurations, encompassing capital expenditure, operating costs, and return on investment to ensure the process design maintains economic competitiveness.

Environmental Sustainability (\mathcal{E}_s) : This metric assesses environmental impact and sustainability, covering pollutant emissions, carbon footprint, and resource recycling to ad-

vance green chemistry objectives.

Process Safety (\mathcal{P}_s): This criterion ensures inherent process safety through evaluation of hazardous material handling, safety system configurations, and accident risk assessment to identify potential hazards.

Technical Feasibility (\mathcal{T}_f): This dimension validates the maturity and implementability of process technologies, evaluating reaction mechanism rationality, separation technology applicability, and operational control complexity.

Topological Rationality (\mathcal{T}_r): This metric evaluates the completeness and optimization degree of process topology, including unit operation connectivity logic and material flow rationality to verify efficient process structure design. The overall evaluation score for each iteration is computed as a weighted combination of the individual dimensions.

$$\mathcal{S}_i = w_1 \cdot \mathcal{E}_f + w_2 \cdot \mathcal{E}_s + w_3 \cdot \mathcal{P}_s + w_4 \cdot \mathcal{T}_f + w_5 \cdot \mathcal{T}_r \quad (1)$$

where \mathcal{S}_i represents the comprehensive evaluation score for the i -th iteration, and the weights for the five evaluation dimensions are 0.35, 0.25, 0.15, 0.15, and 0.10 (Seider et al. 2016), respectively, prioritizing economic and environmental considerations in industrial applications. In case of simulation failure, a penalty factor is applied: $\mathcal{S}_i^{fail} = \lambda \cdot \mathcal{S}_i$, where λ is set to 0.3.

Enhanced MCTS

Inspired by AFlow (Zhang et al. 2024b), our Enhanced Monte Carlo Tree Search (E-MCTS) approach aims to address a critical limitation in iterative search processes, where configurations that fail to converge are often overlooked despite containing valuable design insights worthy of further exploration. In E-MCTS structure, each tree node represents a complete process configuration rather than individual unit operations, enabling the exploration of diverse complex chemical process designs.

Initialization E-MCTS begins with three initial nodes generated by workflow, derived from the user’s input, serving as the foundation for systematic exploration. The root node is created as a virtual node, with the three initial configurations becoming its child nodes. We initialize the dynamic candidate pool as an empty set and establish key parameters, including the dynamic weight $\alpha(0)$ for initial exploration focus and the exploration coefficient $c(0)$.

Selection with Dual-Layer Value Evaluation We propose a dual-layer value evaluation system to address the unique challenge where failed configurations may contain valuable design insights. Our approach distinguishes between immediate value (derived directly from simulation scores) and potential value (considering multi-dimensional metrics). For failed configurations, we recognize exceptional performance in specific dimensions as indicators of refinement potential. The combined value integrates both perspectives through dynamic weighting:

$$V(n_i, t) = \alpha(t) \cdot V_{\text{imm}}(n_i) + (1 - \alpha(t)) \cdot V_{\text{pot}}(n_i) \quad (2)$$

where $\alpha(t)$ transitions from exploration-focused to exploitation-focused as the search progresses. We further

enhance node selection by incorporating domain-specific features, augmenting the standard UCB formula:

$$\text{UCB}_{\text{enh}}(n_i, t) = V_{\text{comb}}(n_i, t) + c(t) \sqrt{\frac{\ln v_p}{v_i}} + \Psi(n_i) \quad (3)$$

where the feature weighting function $\Psi(n_i)$ incorporates recent improvement indicators, variance-based sensitivity measures, and depth preferences.

Expansion During the expansion phase, each selected node generates three child configurations. We execute a new workflow iteration to update configurations, leveraging both the selected node’s configuration and historical insights from past iterations. The experience encompasses all modifications with their corresponding improvements or failures, along with precise logs of predictions and expected outputs, maximizing insight extraction from previous attempts.

Dynamic Revisit Mechanism In contrast to traditional MCTS which abandons low-scoring branches, we maintain a dynamic candidate pool containing the second and third highest-scoring nodes that have not been revisited. When global improvement stagnates, the algorithm selects nodes for revisitation based on the gap between potential and realized performance:

$$n_{\text{rev}} = \arg \max_{n \in \mathcal{R}} (V_{\text{pot}}(n) - V_{\text{imm}}(n)) \quad (4)$$

This mechanism prevents premature convergence and enables the algorithm to escape local optima by revisiting promising configurations that initially failed due to suboptimal parameters.

Terminal Conditions We implement comprehensive termination criteria tailored to chemical process optimization: (1) discovery of configurations exceeding target performance, (2) reaching iteration limits, (3) global improvement stagnation across all active leaves, or (4) high convergence indicated by concentrated visit frequencies and stable scores. The final solution selection combines solution quality (70% weight), search confidence as measured by visit frequency (20% weight), and stability indicators (10% weight), ensuring robust and reliable process configurations.

This E-MCTS algorithm effectively addresses the challenges of chemical process optimization by recognizing value in failed configurations, dynamically balancing exploration and exploitation, and implementing intelligent revisit strategies to systematically navigate the vast design space.

Experiment

Baselines

We evaluate our approach against four categories of baselines: (1) End-to-end methods: We prompt GPT-4o and Claude Sonnet 4 to directly generate JSON request configurations. (2) Multi-agent workflows: We employ Swarm, AutoGen, CrewAI, MetaGPT to benchmark our multi-agent workflow’s effectiveness. (3) Human expert baseline: We invite three chemical engineering experts to perform manual design and computational validation. (4) Our methods: We evaluate two variants of our proposed approach—a base workflow without thermodynamic analysis workflow and an enhanced version incorporating thermodynamic analysis.

Methods	Main Scores						Efficiency	
	$\mathcal{E}_f \uparrow$	$\mathcal{E}_s \uparrow$	$\mathcal{P}_s \uparrow$	$\mathcal{T}_f \uparrow$	$\mathcal{T}_r \uparrow$	$\mathcal{S} \uparrow$	Time(s) \downarrow	SCR(%) \uparrow
GPT-4o(Achiam et al. 2023)	73.6	77.2	71.4	75.5	69.8	34.36	86.13	23.4
Claude-Sonnet-4(Anthropic 2025)	74.3	73.9	76.1	72.7	71.5	33.16	89.24	21.2
Swarm(OpenAI 2024)	79.9	79.4	80.3	79.7	80.1	49.32	149.21	45.4
AutoGen(Wu et al. 2024)	79.4	78.2	80.1	79.6	78.3	48.27	303.16	44.3
CrewAI(crewAIInc 2023)	79.8	77.5	78.1	79.3	79.8	49.95	389.09	47.6
MetaGPT(Hong et al. 2023)	81.2	79.9	80.5	81.0	80.4	51.98	612.33	49.2
Expert manual	87.2	85.6	84.6	86.4	85.9	86.16	8301.91	100.0
Ours-4o	85.7	84.9	83.8	86.4	84.2	73.01	892.04	79.6
Ours*-4o	85.8	86.0	84.3	86.5	85.5	73.88	913.21	80.3

Table 1: Evaluation of our workflow on Simona dataset. ‘‘Ours’’ represents a pure process configuration generation workflow, while ‘‘Ours*’’ represents a configuration generation workflow that introduces thermodynamic analysis workflow

Dataset

Our experiments are conducted on a self-constructed dataset called Simona, consisting of 1000 process descriptions carefully designed by chemical engineering experts. The dataset encompasses process descriptions of varying complexity, primarily distributed across: (1) Unit operations of different difficulty levels (i.e., nodes in a process flowsheet), including mixers, distillation columns, among others. (2) Process descriptions with varying levels of detail—for relatively simple cases, we provide comprehensive process descriptions including component specifications, unit model specifications, and partial parameter specifications; for more challenging cases, we deliberately provide ambiguous process descriptions with certain critical information withheld.

Implementation Details

Our workflow is built upon LangGraph(langchain ai 2024) (v0.4.10) with LangChain (v0.3.26) providing the foundation for LLM interactions, as our workflow involves iterative loops that align well with LangGraph’s characteristics. We employ gpt-4o as the base language model for all agents. The temperature parameters for the task understanding agent and evaluation analysis agent are set to 0.1 to ensure relatively stable task comprehension and result evaluation. In contrast, the temperature parameters for the topology generation agent and parameter configuration agent are set to 0.9 to promote greater diversity and innovation in the process schemes generated across different branches. For chemical process simulation, we integrate with our self-developed chemical process simulation software through HTTP APIs.

Main Results

Main Scores. Table 1(left) presents the evaluation results across five dimensions along with the overall score, which combines scores from both successful and failed simulation cases. Due to our conservative scoring mechanism in the evaluation agent, our method does not demonstrate substantial advantages across individual dimensions compared to baseline approaches. However, our method achieves significantly higher overall scores than other end-to-end and gen-

eral multi-agent methods, primarily attributed to our simulation convergence rate in process simulation software. Notably, expert manual design exhibits clear superiority in economic feasibility, as evidenced by the optimized topology structures and computational results. Our method achieves comparable performance to expert design across other dimensions and ranks second in overall scores. Since scoring alone cannot fully capture the significance of automated design workflows, we further analyze efficiency metrics.

Efficiency. Table 1(right) presents efficiency metrics across different methods. We recruited three chemical engineering PhD students to design processes based on requirement descriptions and recorded their average design time. Regarding Simulation Convergence Rate (SCR)—defined as the ratio of successfully converged process designs to total requirements—expert manual design achieves 100% SCR due to extensive parameter optimization experience. Our method achieves 80.3% SCR, substantially outperforming other baseline methods. In terms of time efficiency, GPT-4o demonstrates remarkable speed, generating process configurations in merely 86.13 seconds on average. However, its SCR of only 23.4% indicates that it can merely handle simple processes with detailed parameter specifications, offering limited practical value for automated process design. While expert manual design achieves exceptional SCR, it requires considerable time investment—8301.91 seconds on average—with complex processes often demanding significantly longer durations. Our method maintains high SCR while achieving 89.0% reduction in design time compared to manual approaches, representing a significant advancement in accelerating chemical process design.

Qualitative Analysis. To intuitively compare output differences across methods, we conduct qualitative analysis as shown in Figure 3. We select representative configuration outputs from different baselines and visualized the corresponding process flowsheets using chemical process simulation software. In the flash drum separation process, the output of GPT-4o exhibits a fundamental flaw: while a flash drum requires one inlet stream and two outlet streams (liquid and vapor phases), the generated configuration includes

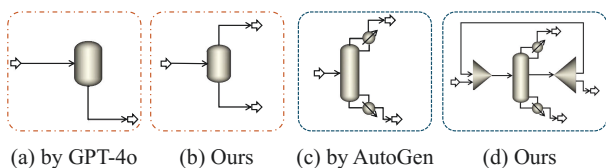


Figure 3: Illustration of the design generated by different methods, where different colors represent different cases.

only a single outlet stream. This represents a common failure resulting from the absence of computational validation checks. For the benzene-toluene separation process, the flowsheet generated by AutoGen meets basic functional requirements but lacks off-specification product recycle loops. Our work, accounting for practical economic constraints, incorporates reflux streams to ensure product quality specifications. While the overall process topologies appear similar, the economic performance differs dramatically, demonstrating the value of our chemical process design agent workflow.

Ablation Study

Analysis of Key Components. We conducted ablation experiments on two critical components: Task Understanding Agent(TUA) and E-MCTS, with results presented in Table 2. Upon removing the Task Understanding Agent, the workflow output scores decreased significantly. This indicates that without systematic context extraction and understanding of user inputs, relying solely on the process design and parameter optimization agents to directly process raw user inputs leads to critical constraint omissions, resulting in degraded generation quality and substantially reduced iteration efficiency. Similarly, disabling E-MCTS resulted in a significant drop in the convergence rate. Our experiments revealed that initial topology and parameter configuration flaws severely impair subsequent iteration efficiency, causing the process to terminate prematurely when score improvements plateau.

Model	$\mathcal{E}_f \uparrow$	$\mathcal{E}_s \uparrow$	$\mathcal{P}_s \uparrow$	$\mathcal{T}_f \uparrow$	$\mathcal{T}_r \uparrow$	Iter \downarrow	SCR \uparrow
Ours*-4o	85.8	86.0	84.3	86.5	85.5	15	80.3
w/o E-MCTS	84.9	85.5	82.9	84.9	84.2	7	74.8
w/o TUA	83.2	83.1	81.0	83.3	82.6	18	78.9

Table 2: Ablation study of the impact of key components on execution efficiency

Impact of E-MCTS Nodes. We performed ablation experiments on the number of child nodes in E-MCT, evaluating settings of 2, 3, 4, and 5 children per expansion. As shown in Table 3, lower child node counts accelerate convergence and reduce token consumption in Simona, yet yield lower SCR due to limited exploration, which constrains the generation of high-quality solutions. Conversely, higher child node counts slow convergence and increase token consumption, but the expanded search space enhances the probability of discovering superior configurations, resulting in improved SCR. To balance convergence efficiency,

token costs, and solution quality, we adopted a child node count of 3 as the optimal trade-off.

E-MCTS Nodes	Time(s) \downarrow	Tokens(K) \downarrow	SCR(%) \uparrow
2	736.55	794.2	79.8
3	913.21	994.7	80.3
4	1102.72	1192.1	80.5
5	1197.09	1325.6	80.6

Table 3: The impact of the number of E-MCTS nodes on time and token efficiency

Analysis of ICL Methods. Table 4 presents ablation studies on in-context learning (ICL) within our workflow. We removed the process-construction chain-of-thought (CoT) prompting from the topology generation agent and few-shot parameter examples from the parameter configuration agent. Experimental results demonstrate that these components are crucial: Their absence compromises both process design rationality and computational convergence, as initial parameter values significantly impact computational efficiency in complex chemical processes.

Model	$\mathcal{E}_f \uparrow$	$\mathcal{E}_s \uparrow$	$\mathcal{P}_s \uparrow$	$\mathcal{T}_f \uparrow$	$\mathcal{T}_r \uparrow$	SCR(%) \uparrow
Ours*-4o	85.8	86.0	84.3	86.5	85.5	80.3
w/o CoT	84.6	84.1	84.0	85.2	84.9	78.8
w/o Few-Shot	85.3	85.7	83.9	85.8	85.2	77.9

Table 4: Ablation study of ICL methods on execution efficiency

Conclusion

In this paper, we have presented a comprehensive study on automated chemical process design and validation that accelerates both the design phase and the simulation-based verification phase, significantly reducing the design time for chemical engineering practitioners. We have introduced a novel automated workflow for generating computational configurations in chemical process simulation software, departing from the traditional paradigm of manual configuration from flowsheets. This establishes a foundational infrastructure for accelerating iterative design validation in chemical process engineering. Our work decomposes the process design validation workflow and leverages E-MCTS to improve design robustness. Experiments on Simona dataset demonstrate the effectiveness of our approach. Compared to existing methods, our work not only generates feasible topological structures but also produces outputs that are computationally verifiable. Furthermore, our research underscores the potential of automated chemical process design. While there remains scope for further refinement compared to expert-designed solutions, the significant advantages in time efficiency establish a practical and extensible methodology for future chemical process design endeavors.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2022YFB3305900), National Natural Science Foundation of China (62394345, 62103151), the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017 and Fundamental Research Funds for the Central Universities (222202517006).

References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Anthropic. 2025. Introducing Claude 4. "https://www.anthropic.com/news/claude-4". Accessed: 2025-6-25.
- Balhorn, L. S.; Gao, Q.; Goldstein, D.; and Schweidtmann, A. M. 2022. Flowsheet recognition using deep convolutional neural networks. In *Computer Aided Chemical Engineering*, volume 49, 1567–1572. Elsevier.
- Boiko, D. A.; MacKnight, R.; Kline, B.; and Gomes, G. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992): 570–578.
- Bran, A. M.; Cox, S.; Schilter, O.; Baldassari, C.; White, A. D.; and Schwaller, P. 2023. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*.
- Cen, J.; Liu, J.; Li, Z.; and Wang, J. 2025. SQLFixAgent: Towards Semantic-Accurate Text-to-SQL Parsing via Consistency-Enhanced Multi-Agent Collaboration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 49–57.
- Chen, W.; Su, Y.; Zuo, J.; Yang, C.; Yuan, C.; Qian, C.; Chan, C.-M.; Qin, Y.; Lu, Y.; Xie, R.; et al. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4): 6.
- Chen, Y.; Feng, J.; Zhang, H.; Gong, L.; Zhu, F.; Zhao, R.; Hou, Q.; Cheng, M.-M.; and Song, Y. 2025. Re-Aligning Language to Visual Objects with an Agentic Workflow. *arXiv preprint arXiv:2503.23508*.
- crewAI Inc. 2023. CrewAI. "https://github.com/crewAIInc/crewAI". Accessed: 2025-6-10.
- d'Anterrosches, L. 2006. *Process flow sheet generation & design through a group contribution approach*. Technical University of Denmark.
- Douglas, J. M. 1988. Conceptual design of chemical processes. (*no Title*).
- Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; et al. 2023. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.
- langchain ai. 2024. LangGraph. "https://github.com/langchain-ai/langgraph". Accessed: 2025-6-10.
- Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023. Camel: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36: 51991–52008.
- Li, Z.; Xu, S.; Mei, K.; Hua, W.; Rama, B.; Raheja, O.; Wang, H.; Zhu, H.; and Zhang, Y. 2024. Autoflow: Automated workflow generation for large language model agents. *arXiv preprint arXiv:2407.12821*.
- Liu, S.; Cheng, H.; Liu, H.; Zhang, H.; Li, F.; Ren, T.; Zou, X.; Yang, J.; Su, H.; Zhu, J.; et al. 2024a. Llava-plus: Learning to use tools for creating multimodal agents. In *European conference on computer vision*, 126–142. Springer.
- Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; Zhang, S.; Deng, X.; Zeng, A.; Du, Z.; Zhang, C.; Shen, S.; Zhang, T.; Su, Y.; Sun, H.; Huang, M.; Dong, Y.; and Tang, J. 2024b. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations*.
- Liu, Z.; Yao, W.; Zhang, J.; Xue, L.; Heinecke, S.; Murthy, R.; Feng, Y.; Chen, Z.; Niebles, J. C.; Arpit, D.; et al. 2023. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents. *arXiv preprint arXiv:2308.05960*.
- Mann, V.; Brito, K.; Gani, R.; and Venkatasubramanian, V. 2022. Hybrid, interpretable machine learning for thermodynamic property estimation using grammar2vec for molecular representation. *Fluid Phase Equilibria*, 561: 113531.
- Mann, V.; Gani, R.; and Venkatasubramanian, V. 2023. Group contribution-based property modeling for chemical product design: A perspective in the AI era. *Fluid Phase Equilibria*, 568: 113734.
- Mann, V.; Sales-Cruz, M.; Gani, R.; and Venkatasubramanian, V. 2024. eSFILES: Intelligent process flowsheet synthesis using process knowledge, symbolic AI, and machine learning. *Computers & Chemical Engineering*, 181: 108505.
- OpenAI. 2024. Swarm. "https://github.com/openai/swarm". Accessed: 2025-6-10.
- Pang, W.; Lin, K. Q.; Jian, X.; He, X.; and Torr, P. 2025. Paper2Poster: Towards Multimodal Poster Automation from Scientific Papers. *arXiv preprint arXiv:2505.21497*.
- Qian, C.; Liu, W.; Liu, H.; Chen, N.; Dang, Y.; Li, J.; Yang, C.; Chen, W.; Su, Y.; Cong, X.; et al. 2023. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.
- Qiao, S.; Fang, R.; Qiu, Z.; Wang, X.; Zhang, N.; Jiang, Y.; Xie, P.; Huang, F.; and Chen, H. 2024. Benchmarking agentic workflow generation. *arXiv preprint arXiv:2410.07869*.
- Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, 8821–8831. Pmlr.
- Seider, W. D.; Lewin, D. R.; Seader, J.; Widagdo, S.; Gani, R.; and Ng, K. M. 2016. *Product and process design principles: synthesis, analysis and evaluation*. John Wiley & Sons.

- Song, C. H.; Wu, J.; Washington, C.; Sadler, B. M.; Chao, W.-L.; and Su, Y. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2998–3009.
- Theisen, M. F.; Flores, K. N.; Balhorn, L. S.; and Schweidtmann, A. M. 2023. Digitization of chemical process flow diagrams using deep convolutional neural networks. *Digital Chemical Engineering*, 6: 100072.
- Tula, A. K.; Babi, D. K.; Bottlaender, J.; Eden, M. R.; and Gani, R. 2017. A computer-aided software-tool for sustainable process synthesis-intensification. *Computers & Chemical Engineering*, 105: 74–95.
- Tula, A. K.; Eden, M. R.; and Gani, R. 2019. Hybrid method and associated tools for synthesis of sustainable process flowsheets. *Computers & Chemical Engineering*, 131: 106572.
- Venkatasubramanian, V. 2019. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, 65(1).
- Vogel, G.; Balhorn, L. S.; and Schweidtmann, A. M. 2023. Learning from flowsheets: A generative transformer model for autocompletion of flowsheets. *Computers & Chemical Engineering*, 171: 108162.
- Vogel, G.; Hirtreiter, E.; Schulze Balhorn, L.; and Schweidtmann, A. M. 2023. SFILES 2.0: an extended text-based flowsheet representation. *Optimization and Engineering*, 24(4): 2911–2933.
- Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.
- Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1): 31–36.
- Wu, C.; Kreidieh, A. R.; Parvate, K.; Vinitzky, E.; and Bayen, A. M. 2021. Flow: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*, 38(2): 1270–1286.
- Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.
- Xu, Y.; Su, H.; Xing, C.; Mi, B.; Liu, Q.; Shi, W.; Hui, B.; Zhou, F.; Liu, Y.; Xie, T.; et al. 2023. Lemur: Harmonizing natural language and code for language agents. *arXiv preprint arXiv:2310.06830*.
- Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.
- Zhang, G.; Yue, Y.; Sun, X.; Wan, G.; Yu, M.; Fang, J.; Wang, K.; Chen, T.; and Cheng, D. 2024a. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*.
- Zhang, J.; Xiang, J.; Yu, Z.; Teng, F.; Chen, X.; Chen, J.; Zhuge, M.; Cheng, X.; Hong, S.; Wang, J.; et al. 2024b. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.
- Zhang, T.; Sahinidis, N. V.; and Siirola, J. J. 2019. Pattern recognition in chemical process flowsheets. *AIChE Journal*, 65(2): 592–603.
- Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.
- Zhuge, M.; Wang, W.; Kirsch, L.; Faccio, F.; Khizbullin, D.; and Schmidhuber, J. 2024a. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*.
- Zhuge, M.; Zhao, C.; Ashley, D.; Wang, W.; Khizbullin, D.; Xiong, Y.; Liu, Z.; Chang, E.; Krishnamoorthi, R.; Tian, Y.; et al. 2024b. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*.