Water Research 293 (2026) 125433

Contents lists available at ScienceDirect

Water Research

journal homepage: www.elsevier.com/locate/watres

EPANET-Agentic: A multi-agent system for natural language-controlled simulations of water distribution networks

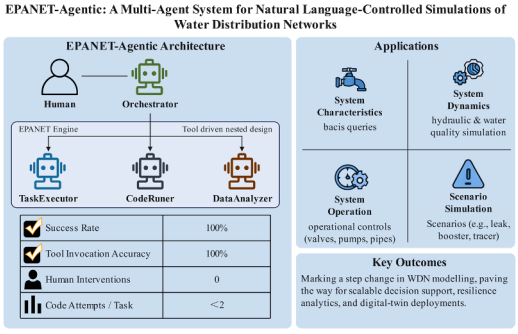
Jian Wang [a], Guangtao Fu [a,*], Dragan Savic [a,b,*]

[a] Centre for Water Systems, University of Exeter, Exeter EX4 4QF, United Kingdom
[b] KWR Water Research Institute, Nieuwegein 3430 BB, the Netherlands

HIGHLIGHTS

- EPANET-Agentic enables autonomous, natural language-driven control of WDN simulations.
- It is a multi-agent system consisting of an orchestrator and three specialised agents.
- It achieves accurate control and full task completion across diverse hydraulic tasks.
- It supports scalable, interpretable, and automation-ready WDN management.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Agentic AI
EPANET
Large language models
Water distribution networks
Workflow automation

ABSTRACT

Water distribution networks (WDNs), a critical part of urban infrastructure, normally require numerous model simulations for effective planning and management. However, traditional WDN modelling requires complex workflows and specialized expertise. EPANET is the most widely adopted modelling tool for WDN hydraulics and water quality simulations, yet its operational complexity restricts accessibility and slows timely decision-making. Recent advances in large language models (LLMs) have led to the development of agentic artificial intelligence systems that autonomously coordinate tasks and control complex engineering simulations through natural language prompts. Here we introduce EPANET-Agentic, a multi-agent system that integrates advanced workflow reasoning with the EPANET simulator and incorporates human-in-the-loop oversight for critical interventions. The new platform adopts an orchestrator-centred, tool-driven architecture that nests three specialised agents (TaskExecutor, CodeRunner, and DataAnalyzer) as function-call tools. This design enables autonomous task decomposition, precise tool invocation, and transparent workflow management. The abilities of EPANET-Agentic are evaluated on three benchmark networks (i.e., L-Town, C-Town, and Net3) across four categories of tasks: System Characteristics, System Dynamics, System Operation, and Scenario Simulation. The results demonstrate that EPANET-Agentic achieved a 100% success rate and tool invocation accuracy with no human interventions. Moreover, the multimodal DataAnalyzer agent provided valid interpretations of simulation results, while the nested tool design ensured robustness and the architecture exhibited strong scalability across diverse hydraulic analysis tasks. These findings confirm that EPANET-Agentic enables natural language-controlled WDN simulation and analysis with engineering-grade reliability, while still adhering to a human-in-the-loop approach

* Corresponding authors.
E-mail addresses: g.fu@exeter.ac.uk (G. Fu), d.savic@exeter.ac.uk (D. Savic).

https://doi.org/10.1016/j.watres.2026.125433
Received 8 October 2025; Received in revised form 10 January 2026; Accepted 20 January 2026
Available online 20 January 2026

required for safety-critical systems. With its modular architecture and strong adaptability, EPANET-Agentic marks a step change from conventional WDN modelling approaches, positioning itself as a next-generation platform for complex planning and management challenges.

1. Introduction

Water distribution networks (WDNs) are critical infrastructure that ensure the continuous and safe water supply to urban populations (Bilal et al., 2021; Marques et al., 2018; Tsiami et al., 2025). Their management faces increasing challenges due to aging infrastructure, population growth, and the need to enhance system resilience in response to future uncertainties (Marques et al., 2018; Safitri et al., 2023). To support effective decision-making, WDN management relies heavily on hydraulic simulation tools, which play essential roles in system planning (Khedr and Tolson, 2016), operational optimization (Marzouny and Dziedzic, 2024), and resilience assessment (Klise et al., 2017). Among these, EPANET is the most widely used platform (Rossman et al., 2020), capable of simulating WDN dynamics of pressure, flow, and water quality, with its extensions (e.g., the open-source Python-based Water Network Tool for Resilience library, WNTR) enabling advanced modelling and resilience analysis (Klise et al., 2020). However, the widespread adoption of these computational tools is constrained by multiple factors, including substantial time requirements for model construction, calibration, and maintenance, as well as their inherent complexity and dependence on specialized expertise. These constraints limit practical use among non-expert operators and slow decision-making in real-world applications (Goldshtein et al., 2025; Sela et al., 2025).

Recent advances in large language models (LLMs) have opened new opportunities for intuitive and natural interactions with complex computational systems. Models such as GPT-5 (OpenAI, 2025) and DeepSeek-V3 (DeepSeek-AI, 2024) demonstrate unprecedented capabilities in understanding and generating human-like text, enabling seamless communication between users and sophisticated analytical systems. Leveraging these developments, a new generation of agentic artificial intelligence (AI) systems has emerged, in which the LLM shifts from being a passive text generator to the central planner of an autonomous reasoning-and-action workflow. Unlike standalone prompting, an agentic AI system enhances the LLM with planning, memory, tool use, and action modules, allowing it to interpret a user task, generate a plan, invoke registered tools (e.g., a simulation model) to interact with the environment, evaluate the returned results, and iteratively refine its steps until the task is completed (Acharya et al., 2025; Fu, 2025; Wang et al., 2024). By enabling natural language control of complex workflows, agentic AI systems are being trialled in various domains. For example, OpenFOAMGPT automates computational fluid dynamics simulations through dialogue-driven interaction (Feng et al., 2025), Paper2Code converts machine learning papers into executable code through a structured multi-agent system (Seo et al., 2025), and Magentic-One coordinates sub-agents via an orchestrator to dynamically solve complex tasks with robust and scalable performance (Fourney et al., 2024).

Despite these advancements, applications of LLMs in WDN management are still scarce, with only a few exploratory studies reported to date. Sela et al. (2025) demonstrated the use of generative AI to support water utility operations through natural language interaction. Taormina and van der Werf (2024) applied large multimodal models to sewer defect detection, showing improved interpretability in predictions. In parallel, Lyu et al. (2025) evaluated large multimodal models for urban floodwater depth estimation, highlighting the potential of GPT-4 to outperform supervised baselines in image-driven flood monitoring. Marzouny and Dziedzic (2024) introduced an LLM-assisted framework for pump operation optimization, where ChatGPT interacted iteratively with EPANET to achieve energy savings surpassing those of genetic algorithms. However, these studies mostly treat LLMs as conversational assistants for single tasks, whereas water utilities require integrated multi-agent systems capable of supporting diverse model-based decisions rather than isolated operations. Thus, they represent preliminary rather than fully realized agentic systems. In a step forward, Goldshtein et al. (2025) introduced an LLM-EPANET framework, which uses a retrieval-augmented pipeline to enable natural language interaction with hydraulic models, highlighting the potential of LLMs to improve decision-making in WDN management. Similarly, Wang et al. (2026) tested a two-agent dialogue framework on model calibration and pump scheduling tasks, demonstrating promising prospects for autonomous reasoning, simulation tool interaction, and code generation. However, a critical knowledge gap remains in applying large language models (LLMs) to automatically transform data into actionable knowledge for informed decision-making—a key element of data-centric water engineering (Fu et al., 2024).

Nevertheless, approaches that rely solely on a single agent or simple dual-agent interactions remain inadequate for enabling water utilities to fully integrate agentic AI into simulation-driven workflows. Such single-task approaches fall short of meeting critical requirements for hydraulic modelling, including numerical accuracy, robustness, reproducibility, and traceability (Wang et al., 2025). Integrating multi-agent systems with EPANET-based WDN simulations remains particularly challenging, as it requires accurate task interpretation (e.g., design, control, or network model calibration), effective decomposition into subtasks, coordinated multi-agent interactions, precise external tool calls, and reliable analysis of simulation outputs (Xi et al., 2025). Furthermore, ensuring operational robustness and mitigating hallucination risks remain unresolved, highlighting both the research gap and the need for a multi-agent system that combines advanced LLM reasoning with structured and dependable hydraulic simulation workflows.

This study proposes EPANET-Agentic, an early-stage LLM-based multi-agent system that is targeted primarily at practitioners such as system operators, field technicians, and practising engineers who perform routine EPANET analyses, by lowering the modelling and operational learning curve. It replaces conventional EPANET interfaces and ad-hoc scripts with a natural-language-driven control layer that integrates LLM-based reasoning and planning with EPANET's robust simulation environment. This system features an Orchestrator-centred architecture with three sub-agents (TaskExecutor, CodeRunner, and DataAnalyzer) embedded as function-calling tools using a tool-driven nested design approach. When a subtask is delegated, the Orchestrator triggers a function that encapsulates the corresponding agent's logic, enabling flexible invocation and management within a tool-driven nested design. A human-in-the-loop mechanism allows users to approve, interrupt, or modify tasks at any stage to retain control over the workflow. Comprehensive experiments were conducted on three benchmark networks (L-Town, C-Town, and Net3), covering four categories of tasks: System Characteristic, System Dynamics, System Operation, and Scenario Simulation. extensive evaluations confirm its reliable performance without requiring active human intervention, while still allowing user oversight and intervention through a human-in-the-loop mode. This balance establishes a foundation for integrating multi-agent systems into structured engineering workflows and supports the transition toward autonomous WDN management.

2. Methodology

EPANET, developed by the U.S. Environmental Protection Agency, is an open-source hydraulic simulation tool widely used to model pressure,

flow, and water quality dynamics in WDNs (Rossman et al., 2020). Its open architecture and proven reliability have fostered the development of the Python-based WNTR library, which enables advanced network modelling and resilience analysis (Klise et al., 2020). Agentic AI systems represent an emerging paradigm that integrates LLMs with autonomous decision-making, planning, and tool execution. Within this system, multiple agents collaborate to manage tasks such as workflow planning, simulation control, and result analysis with minimal human intervention (Ghafarollahi and Buehler, 2025; Wang et al., 2024). In this study, we integrate the WNTR library with a multi-agent system to enable natural language interaction with EPANET for tasks including WDNs control, hydraulic and water quality simulation, and result analysis. Section 2.1 presents the architecture and workflow of the proposed EPANET-Agentic, while Section 2.2 describes the evaluation protocol, including task design, performance metrics, and validation procedures.

2.1. *EPANET-Agentic: Architecture*

The proposed EPANET-Agentic architecture, shown in Fig. 1, employs a team of agents that collaborate to accomplish complex tasks in the context of WDN simulation and analysis. These agents are powered by state-of-the-art general-purpose LLMs - DeepSeek (DeepSeek-AI et al., 2024) and Qwen (Yang et al., 2025) - accessed via their Application Programming Interfaces (APIs). These models were selected to support advanced reasoning and code generation, with multimodal capability used where required. Selection was also guided by a favourable cost-performance balance under budget and local-deployment constraints. DeepSeek served as the primary model, whereas Qwen-VL was used for image and figure analysis. This hybrid stack is therefore well suited to reasoning-intensive and tool-oriented WDN workflows. Each agent is characterized by a unique profile defining its role in the system, as summarized in Table 1, while their detailed profile configurations are provided in Figures S1~S4. Within this dynamic environment, the agents work in coordination to execute simulation workflows efficiently and with limited human intervention:

Table 1
LLM-powered agents implemented in the current study to solve the tasks in WDNs.

Agent # (LLM)	Rationale for LLM selection	Agent name	Agent role
None	None	User	Human-in-the-loop reviewer for plan approval and intervention.
DeepSeek V3	Strong multi-step reasoning and workflow orchestration	Orchestrator	Interprets user queries, generates step-by-step execution plans, and coordinates the actions of sub-agents.
DeepSeek V3	Robust tool/function calling for deterministic execution	TaskExecutor	Calls external tools to validate .inp files, apply controls, and run scenario simulations.
DeepSeek R1	High coding proficiency with logical reasoning	CodeRuner	Generates and executes Python code to run hydraulic or quality simulations, saving and returning results.
Qwen-vl-max	Multimodal capability for visual evidence interpretation	DataAnalyzer	Analyses simulation outputs and provides natural language insights.

Orchestrator: This core agent, powered by DeepSeek-V3 (DeepSeek-AI et al., 2024) (Table 1) and chosen for its superior long-horizon task planning and reliable function/tool calling in multi-step workflows, serves as the central reasoning module of the EPANET-Agentic system. The Orchestrator interprets the user query, analyses its intent, and decomposes it into a sequence of structured sub-tasks, each aligned with a specific operational requirement of the hydraulic modelling workflow. These sub-tasks are then delegated to the appropriate sub-agent through LLM function calls, enabling a modular and hierarchical execution process. In practice, the Orchestrator (i) identifies which sub-agent is responsible for the required operation (e.g.,

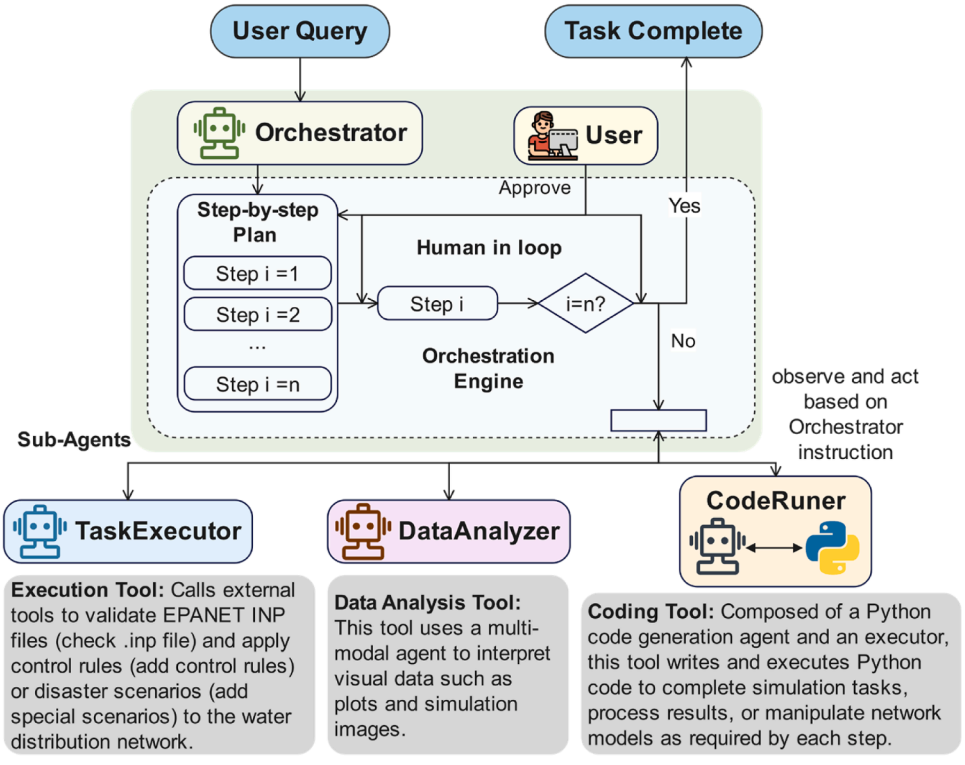


Fig. 1. Architecture and workflow of the EPANET-Agentic for natural language-controlled WDN simulation and analysis.

model manipulation, code execution, or data analysis), (ii) provides the necessary input parameters, and (iii) integrates the returned outputs to update its plan and guide subsequent reasoning steps. A human-in-the-loop mechanism ensures transparency by requiring explicit user approval for each Orchestrator action, such as execution of sub-tasks, receiving results, or deciding whether to continue, requires explicit user approval before proceeding.

TaskExecutor (sub-agent): Equipped with a set of registered tools (Table S1) and powered by DeepSeek-V3 (DeepSeek-AI et al., 2024) (Table 1), selected for its excellent tool-calling reliability and precise function use, this agent (1) validates whether the input EPANET .inp file is correctly formatted and executable, including verifying the existence of nodes, pipes, and other elements mentioned in the user's task description, (2) applies control operations to links (e.g., pipes, pumps, and valves) within the .inp file to perform specified actions, and (3) simulates six special scenarios, including leakage, pollution contamination, booster source injections, fire-fighting demand, and two additional conditions (power outage and earthquake) as provided by WNTR. After execution, the results generated by the tools are returned to the Orchestrator for further processing.

CodeRunner (sub-agent): Powered by DeepSeek R1 (DeepSeek-AI et al., 2025) (Table 1), which excels in code generation, this agent uses a local Python execution module to perform hydraulic and water quality simulations while saving results. When the Orchestrator issues a task that requires CodeRunner, this agent will generate Python code, verify it through local execution, save the outputs as .txt or .png files, and return the results.

DataAnalyzer (sub-agent): Employing the multimodal LLM Qwen-VL-Max (Yang et al., 2025) (Table 1), selected for its superior multimodal reasoning and chart/table understanding, this agent interprets simulation outputs. It analyses textual and visual results from CodeRunner, extracts key insights, summarizes findings, and assists users in interpreting simulation outcomes through natural language.

This modular design ensures reliable execution of complex workflows, including hydraulic and water quality simulation, control application, and performance assessment with clear oversight. By leveraging complementary LLMs capabilities in reasoning and coding, the system reduces manual effort while preserving accuracy and transparency. This approach not only improves the precision and efficiency of WDN modelling but also provides a scalable solution for integrating multi-agent systems into structured WDNs management.

To coordinate these components, the architecture employs a tool-driven nested design in which each sub-agent is exposed as a callable tool with a unique name and typed input schema (Fig. 2). At run time, the Orchestrator selects the next sub-agent and invokes it with a function call that supplies only the parameters required by that agent's interface; the callee executes its encapsulated logic and returns structured outputs to the Orchestrator. For example, when delegating a sub-task to TaskExecutor, the Orchestrator uses the LLM's reasoning to derive a concise sub-task description and an output save path, binds them to the tool's schema as *message* and *path*, and dispatches the call, illustrated by the red-boxed pseudo-payload "*task = f"task: {message}\npath of the file: {path}"*", thereby triggering the TaskExecutor tool. TaskExecutor then validates the inputs and returns a normalized result to the Orchestrator. Consequently, the orchestration remains flexible yet governed by typed contracts, and can lay the foundation for reproducible and auditable experimentation.

2.2. Experimental settings

To evaluate the EPANET-Agentic system, we designed four categories of natural language tasks to assess different aspects of WDN interaction. For clarity, representative examples of the actual questions used in each task category are provided below (with minor variations across different networks), while the complete task description used in our experiments is available in the source code linked in the Data Availability section:

(a) **System characteristics** tasks, which assess basic network properties such as retrieving network components, filtering elements by attributes, and computing key topological metrics, for example, "*how many junctions, tanks, reservoirs, pipes, pumps, and valves are in this WDN*"

(b) **System dynamics** tasks, which evaluate the ability to simulate hydraulic and water quality behaviour, analyse key parameters under baseline and modified conditions, and assess flow-dependent responses, such as "*run a hydraulic simulation, and find the maximum pressure along with its location and time*"

(c) **System operation** tasks, which test the capability to apply and simulate control rules, including time-based, value-based, combined conditions, and multiple rule interactions, to manage network elements during simulations, for example, "*close link 1 at*

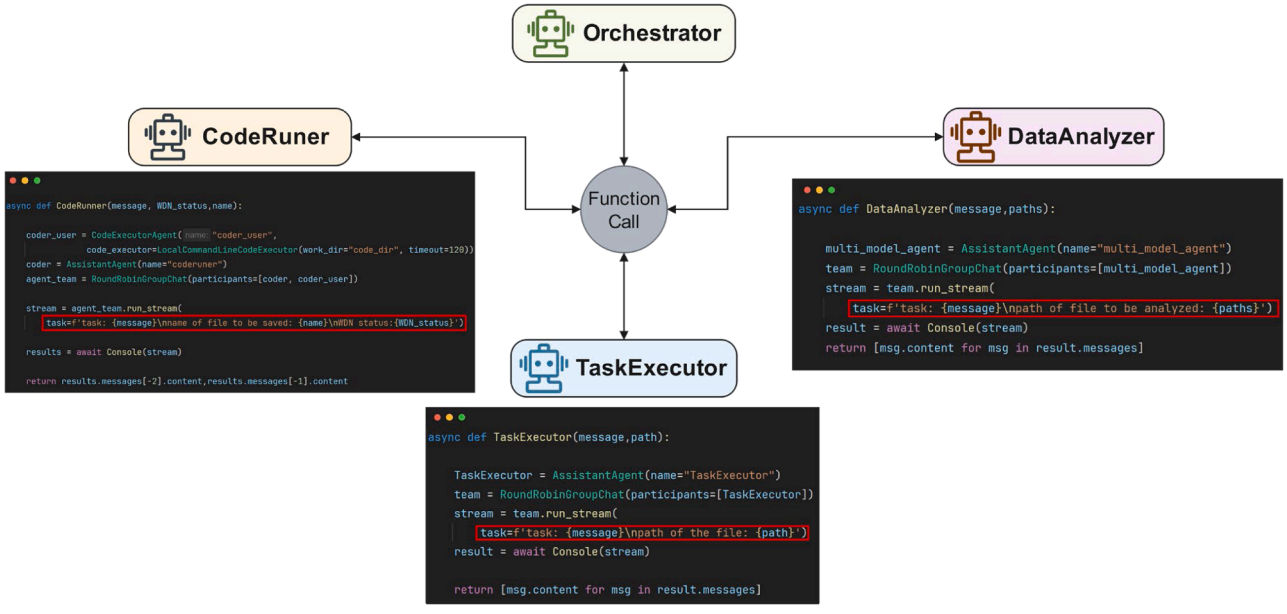


Fig. 2. Function-Call Based Interface Between the Orchestrator and Sub-Agents (schematic pseudo-code for exposition only).

24 h, run a hydraulic simulation, and plot the status time series of link 1″

(d) **Scenario simulation** tasks, which focus on modelling and analysing emergency and operational events in WDNs, including leaks, chemical booster injections, contamination incidents, pump power outages, fire demands, and earthquake impacts, while generating corresponding hydraulic or water quality responses, such as "*simulate a leak scenario at pipe 1 with a leak area of 0.05 m^2 occurring from 24 h to 48 h, run the hydraulic simulation, and plot the time series of flow rate through pipe 1″*

The evaluation of EPANET-Agentic is conducted through a comprehensive assessment of its performance during task-oriented dialogues, covering four key aspects: (1) **Success Rate**: The proportion of tasks for which the system provides the correct final answer, with or without human intervention (e.g., if the system answers 8 out of 10 tasks correctly, the success rate is 80%); (2) **Tool Invocation Accuracy**: The proportion of tool calls that are correctly formatted and semantically appropriate. A tool invocation is considered accurate when the system selects the correct tool and supplies the proper parameters and syntax (e.g., if only 1 of 30 tool calls fails, the accuracy is 96.7%); (3) **Average Code Generation Attempts**: The mean number of code generation/revision cycles required by CodeRunner per task to produce the correct output (e.g., if two tasks require 2 and 1 attempts respectively, the average is 1.5); and (4) **Average Human Intervention Count**: The average number of additional user inputs needed beyond routine approval, such as clarifications or manual corrections (e.g., if one task requires 2 interventions and another requires 1, the average is 1.5).

3. Case study

To evaluate the EPANET-Agentic, three benchmark WDNs were selected, including L-Town, C-Town, and Net3, as illustrated in Fig. 3. L-Town is sourced from Vrachimis et al. (2022), C-Town originates from University of Exeter (2026), and Net3 is provided within the EPANET software package. The key structural attributes of each network are summarized in Table 2. Among these, L-Town is the most complex, consisting of 782 nodes, a total simulation duration of approximately seven days, and a hydraulic time step of five minutes. C-Town is moderately complex with 388 nodes and a 24-hour simulation duration. Net3, the simplest network, includes only 92 nodes, no valve controls, and a 24-hour simulation period. All networks incorporate predefined nodal demand patterns and pump or valve control rules, but do not involve water quality simulations.

4. Results

This section reports results for EPANET-Agentic across four task categories, and the correctness of all resulting outputs, including generated code, tool calls, intermediate artefacts, and final results, was manually verified, with full execution traces and results saved as markdown documents and released publicly at https://mega.nz/file/OhADRKQY#sX8PS7OOrrkQg4Eoq3TPG-dH93JWax33e3HaasmFeFch ttps://github.com/wangjian169/EPANET-Agentic. It proceeds from System Characteristics (Section 4.1), which query static structural properties of the WDN, through System Dynamics (Section 4.2), which evaluate time-varying hydraulic and water-quality behaviour with emphasis on parameter-driven effects, to System Operation (Section 4.3), which integrates operational control rules with simulations, and finally to Scenario Simulation (Section 4.4), which covers representative events including leaks, contamination, and power outages, etc. Section 4.5 summarises performance across categories, quantifying metrics such as task success rate, tool-invocation accuracy, mean code-generation/revision attempts, and the number of human interventions.

4.1. System characteristics

Querying System Characteristics is essential in WDN analysis, as these properties define the network's structure and underpin hydraulic modelling and operational decisions. For example, evaluating network reliability requires determining parameters such as node connectivity, pipe length, and overall topology (Sirsant and Reddy, 2020). Conventional methods often require manual inspection of large-scale models, custom scripting, and extensive data processing, which becomes increasingly cumbersome in complex WDNs. In this section, the EPANET-Agentic automates the extraction and interpretation of network characteristics, supporting both theoretical modelling and practical applications, as depicted in Fig. 4.

Three tasks of increasing complexity were designed: a basic structural query, attribute-based filtering, and topological metric computation, with the complete conversation workflow for each task saved as markdown documents within the source code project. The Orchestrator interprets each user query and generates a stepwise execution plan. Each task follows a consistent workflow: (1) TaskExecutor calls 'check .inp file' tool to verify the .inp file and confirm the existence of task-relevant elements [e.g., whether node 'n1' exists in L-Town for task (c)]; (2) CodeRunner generates and executes Python scripts (all manually reviewed and confirmed correct), saving results in .txt files; (3) DataAnalyzer is optionally invoked to interpret and summarise outcomes. Results from task (aTa) align with the reference data in Table 2, with values such as 782 junctions and 905 pipes matching the corresponding entries, confirming the EPANET-Agentic's accuracy in retrieving basic network components. Task (b) successfully identifies the 10 longest pipes along with their IDs and lengths, and task (c) computes the topological metrics for node 'n1' (degree = 1, eccentricity = 76, and both betweenness and closeness centrality equal to 0), both achieved through

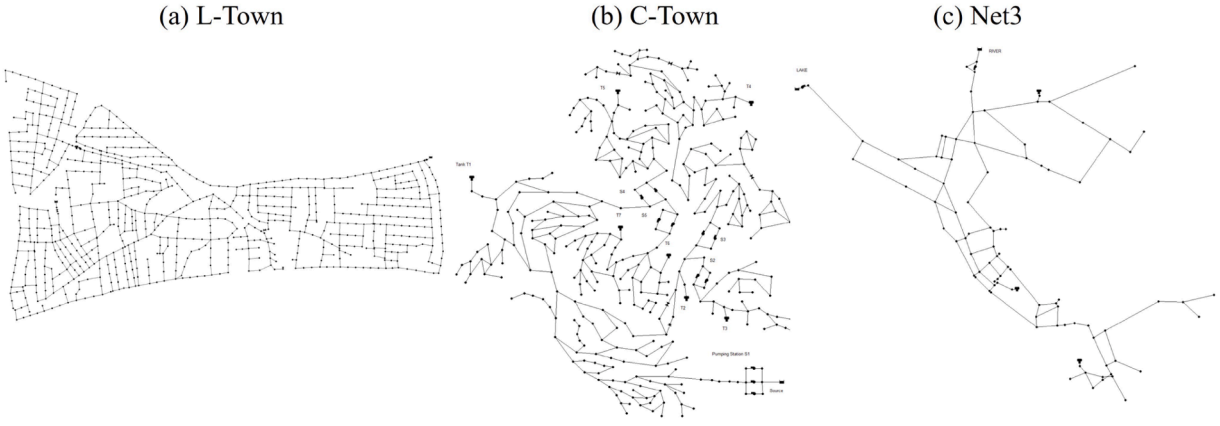


Fig. 3. Topological layouts of the three benchmark WDNs used in this study: (a) L-Town, (b) C-Town, and (c) Net3.

Table 2
Structural characteristics of the evaluated WDNs (L-Town, C-Town, Net3).

WDNs	Reservoirs	Junctions	Tanks	Pipes	Pumps	Valves	Total Durations (h)	Time Step (min)
L-Town	2	782	1	905	1	3	168	5
C-Town	1	388	7	429	11	4	24	5
Net3	2	92	3	117	2	0	24	60

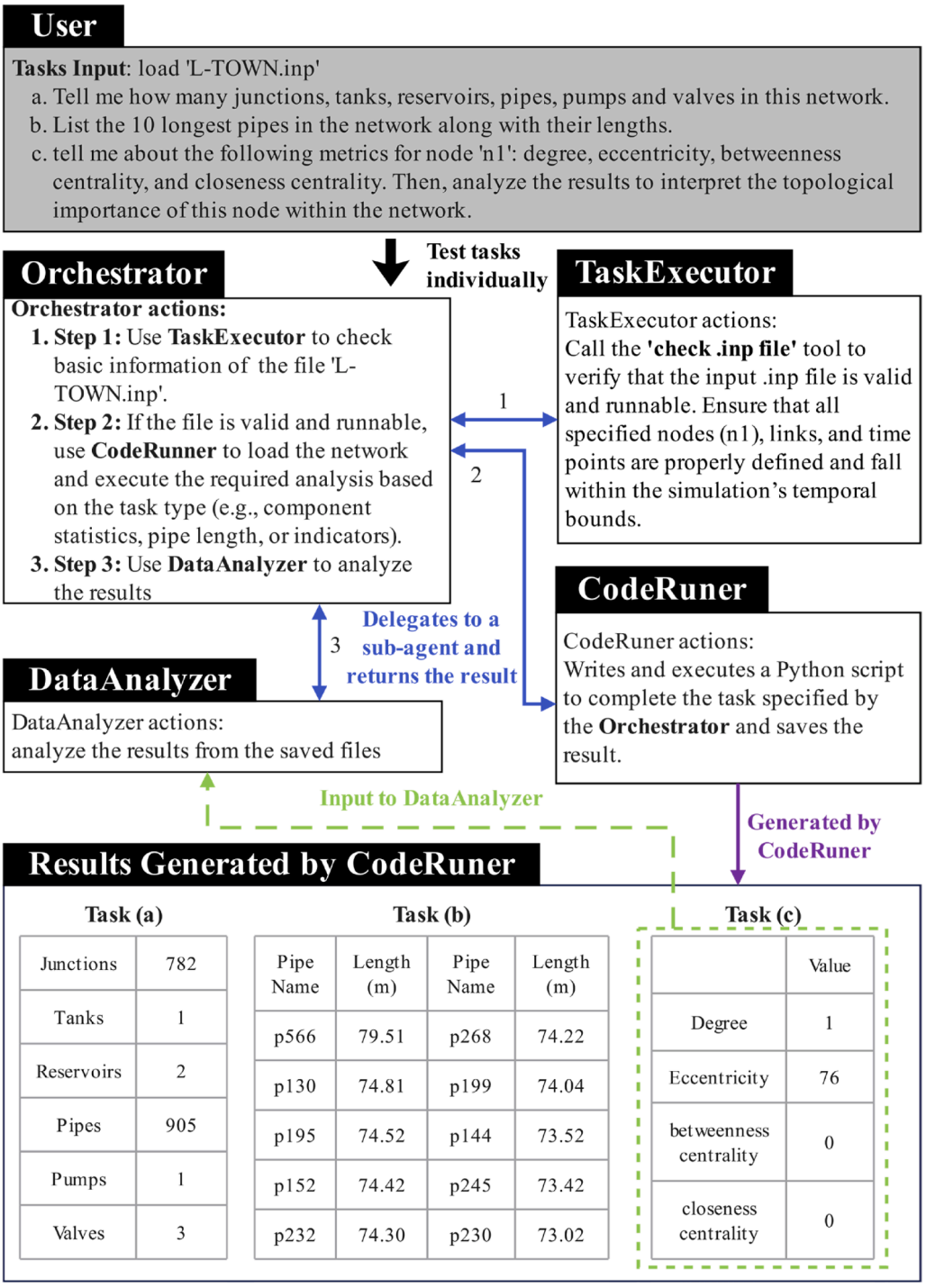


Fig. 4. Three illustrative System Characteristics tasks using EPANET-Agentic: (a) component counts, (b) attribute-based filtering, and (c) topological metrics.

Python code generated and executed by CodeRunner (details in the source code). These values indicate that node 'n1' is peripherally located with minimal influence on network connectivity, a conclusion corroborated by DataAnalyzer's interpretation (see detailed description in Figure S5). Together, these results confirm the ability of EPANET-Agentic to accurately extract, compute, and reason about structural

properties from natural language queries.

4.2. System dynamics

Building on static structural queries, this experiment evaluates the EPANET-Agentic's ability to simulate dynamic hydraulic and water quality behaviour, particularly in response to parameter changes. Parameters such as pipe diameter have a significant influence on pressure distribution patterns and network performance, as widely demonstrated in hydraulic modelling studies (Monsef et al., 2019). To evaluate these effects, pressure distribution plots generated under different simulation scenarios are analysed to reveal how structural modifications alter system dynamics. This experiment leverages the multimodal reasoning capability of the DataAnalyzer agent to interpret visual simulation outputs and extract insights.

As illustrated in Fig. 5, three tasks were designed: hydraulic simulation, water quality simulation, and assessing the impact of pipe diameter changes on pressure distribution (see Figure S6 for enlarged plots). The Orchestrator generates a multi-step plan similar to that in Section 4.1. After verifying the .inp file and task-specific constraints (e.g., simulation duration, existence of node 'R1'), CodeRunner implements the main procedures through automated Python scripting (all

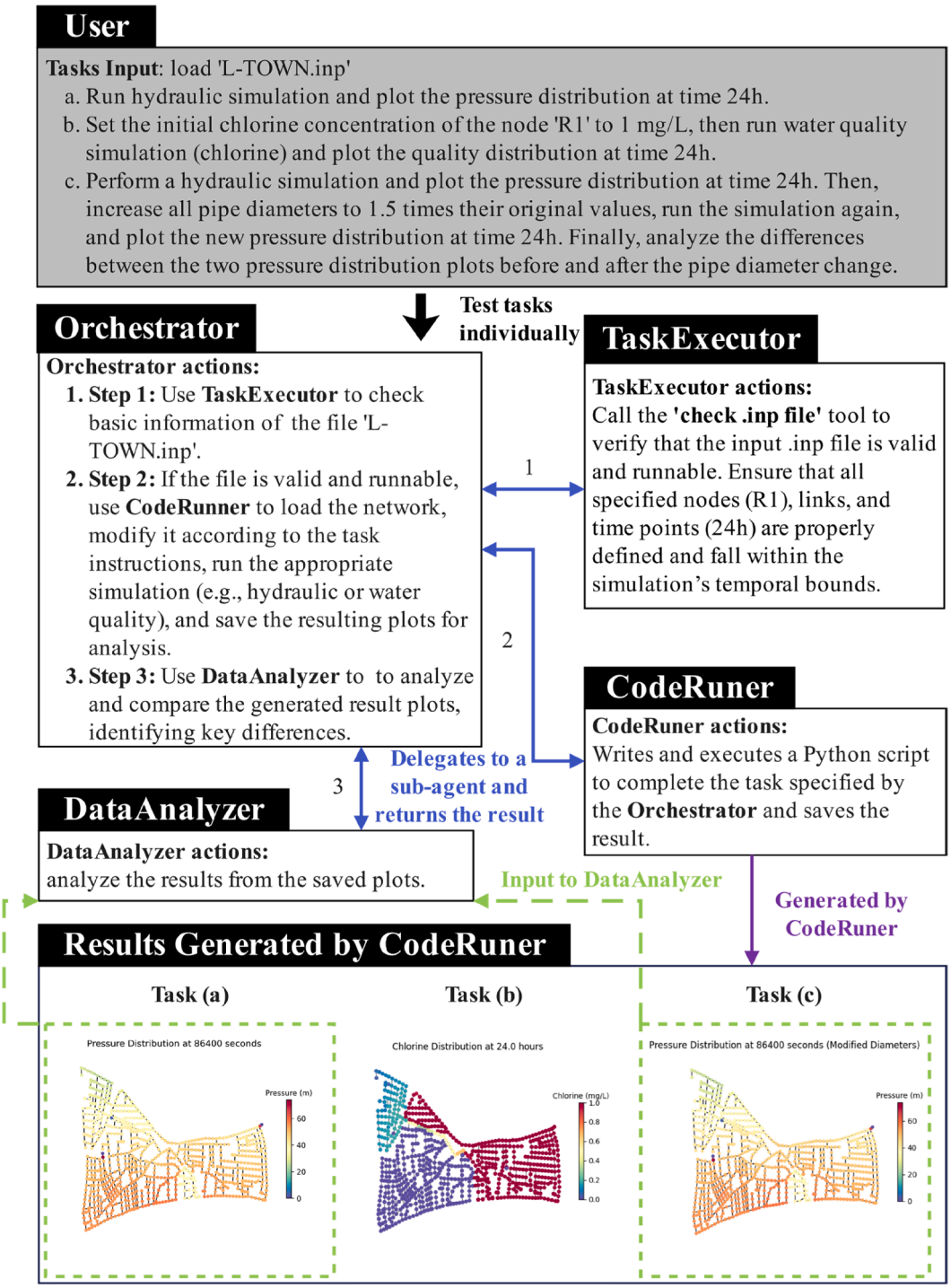


Fig. 5. Three illustrative System Dynamics tasks using EPANET-Agentic: (a) hydraulics, (b) water quality, and the (c) impact of diameter changes.

manually reviewed and confirmed correct). The final and critical step of this experiment involves comparing pressure distributions before and after the diameter change, a task accomplished by the multimodal agent DataAnalyzer. Through image reasoning, DataAnalyzer successfully identified variations in colour scales, pressure ranges, and spatial distribution patterns between the two plots. It concludes that the change improves uniformity and reduces pressure imbalances, a subtle distinction challenging for human observers due to high visual similarity (see detailed description in Figure S7). To support this conclusion, the pressure coefficient of variation was also computed before and after the modification, decreasing from 0.18093 to 0.18073, which confirms a slight improvement in uniformity. This demonstrates the agent's superior sensitivity in detecting nuanced hydraulic changes compared to manual inspection.

4.3. System operation

This experiment examines the system's capacity to simulate operational controls, such as time-based and condition-driven valve actions, and evaluate their effects on hydraulic behaviour. Control rules in WDN models are defined by specifying actions (e.g., closing a valve) triggered by conditions (e.g., tank pressure thresholds), which are then combined and applied to the network to regulate its operation (Klise et al., 2020).

As shown in Fig. 6, three complex tasks were designed: (a) time-based control, (b) multi-condition threshold control, and (c) multi-rule control (see Figure S8 for details). The Orchestrator decomposes each query into a structured plan. After checking the initial input file and task-specific constraints, the TaskExecutor calls the appropriate tools to add the corresponding control rules to the network, and CodeRunner executes the generated Python script to run the simulation and produce the output plots (all manually reviewed and confirmed correct). In task (a), pressure at node 'n111' remains stable until 24 h, when periodic fluctuations appear, confirming the closure of valve 'PRV-2'. Task (b) shows a pronounced pressure shift after 27 h, indicating successful activation of multi-condition control. In task (c), multiple rules induce

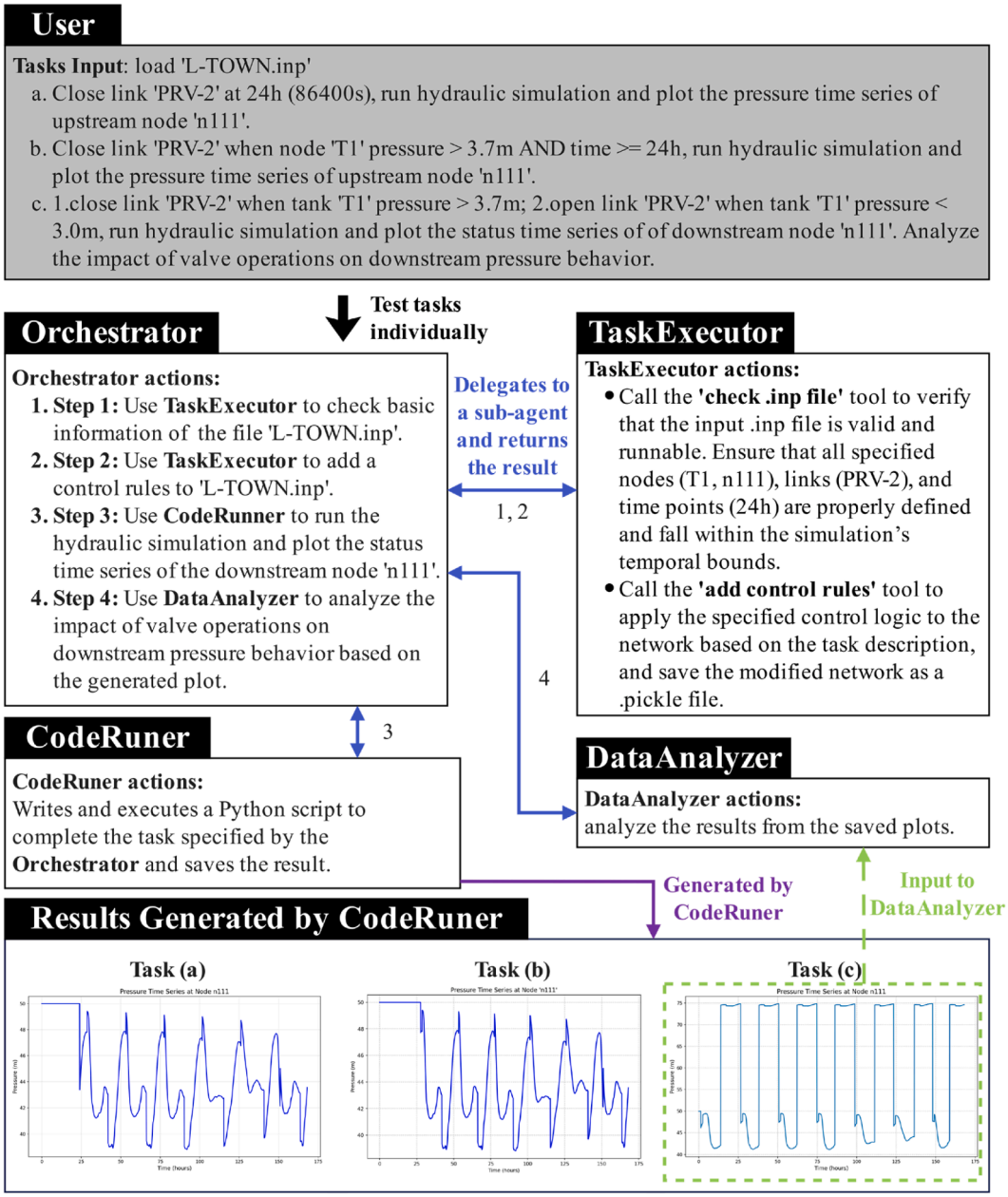


Fig. 6. Three illustrative System Operation tasks using EPANET-Agentic: (a) time-based, (b) multi-condition, and (c) multi-rule controls.

cyclic pressure variations, which DataAnalyzer interprets as a response to tank 'T1' pressure triggering valve 'PRV-2' operations. The DataAnalyzer accurately identified peak pressures (~75 m) and troughs (~42 m) (see detailed description in Figure S9), demonstrating its ability to explain how control logic dynamically influences system behaviour. These results confirm that EPANET-Agentic reliably handles complex control simulations, integrates operational logic with automated execution, and provides interpretable insights into dynamic network responses.

4.4. Scenarios simulation

This experiment evaluates the system's performance in simulating special scenarios, including leaks, contamination, booster injections, power outages, fires, and earthquakes, using the versatile modelling capabilities of the WNTR library. Each scenario requires specific parameters, as outlines in Table 3. The system incorporates safeguards that return errors if essential parameters are omitted.

Previous experiments have extensively evaluated the capabilities of DataAnalyzer across various scenarios, including textual analysis, graphical interpretation, and comparative reasoning; therefore, this section concentrates on the outcomes of special scenario simulations, with enlarged versions of the results provided in Figure S10 for clarity. The Orchestrator coordinates a three-step process: TaskExecutor validates the .inp file, and configures scenario parameters, CodeRunner executed simulations, and results were saved as visual outputs. As shown in Fig. 7, all scenarios produced hydraulically and qualitatively plausible results: leak-induced flow increases in pipes 'p1' and 'p2' between 24 h and 48 h; a chlorine concentration spike downstream node 'n352' of injection node 'n2'; a distinct tracer peak at node 'T1' during contamination; frequent on/off cycling of pump 'PUMP_1' between 24 h and 48 h due to interactions between the power outage control and existing operational rules; increased flow due to fire demand at node 'n1'; and spatially varied Peak Ground Acceleration (PGA) acceleration from earthquake simulation. These outcomes confirm that EPANET-Agentic effectively automates the configuration, execution, and interpretation of diverse emergency scenarios, highlighting its utility in resilience-oriented modelling.

4.5. Trustworthiness verification

High reliability is a critical requirement for EPANET-Agentic, as even minor inconsistencies can lead to significant deviations in results. A well-designed EPANET-Agentic system must therefore demonstrate excellent reproducibility to ensure its applicability in real-world scenarios. To rigorously evaluate this aspect, EPANET-Agentic is assessed using the four evaluation criteria defined in Section 2.2. Based on these criteria, extensive validation experiments were conducted across all test cases, as summarised in Table 4.

To assess reliability and reproducibility, the system was tested across 39 tasks spanning all categories (11 System Characteristics, 11 System Dynamics, 8 System Operation, and 9 Scenario Simulation tasks) over the three networks (with the complete dialogue traces for all tests preserved in the source code project). As summarized in Table 4, the system achieved 100% success rates and tool-invocation accuracy without human intervention beyond routine approval. The average number of code generation attempts per task remained below 2, indicating that CodeRunner typically valid results after one correction at most. System Characteristics tasks required the fewest attempts (as low as 0.9 for C-Town), since initial .inp file check often sufficed. System Dynamics tasks required more iterative coding, with C-Town peaking at 1.7 attempts, as CodeRunner had to directly generate and execute scripts for dynamic simulations, while in Operation and Scenario Simulation tasks auxiliary tools managed control insertion or scenario configuration, which reduced coding demands and kept attempt counts lower. These results confirm the operational robustness and practicality of EPANET-Agentic for real-world WDN management.

Table 3
Parameters required for simulating different special scenarios in EPANET-Agentic.

Scenarios	Necessary parameters
Leak	Leak Pipe Name, Leak Area (m^2), Start Time, End Time
Booster Source Scenarios for water quality	Booster location (Node Name), Source type, Baseline source strength, Start Time, End Time
Contamination	Contaminant Location (Node Name)
Power Outage	Power-Outage Component Name, Start Time, End Time
Fire	Fire Location (Node Name), Fire Demand, Start Time, End Time
Earthquake	Epicenter Coordinates, Magnitude, Depth, Output (PGA or PGV)

4.6. Sensitivity and hallucination evaluation results

To evaluate the sensitivity of EPANET-Agentic to variations in task descriptions, a Semantic Stress Test was conducted using task (b) from Section 4.4 as the **Base Case**. Three variations were designed: (1) **No punctuation**, "*simulate a chemical booster scenario at node n2 using the SETPOINT method with a strength of 1000 and an activation pattern from 24 h to 48 h run the water quality simulation and plot the time series of quality through node n352*"; (2) **Missing sentence**, "*simulate a chemical booster scenario at node n2 using the SETPOINT method with a strength of 1000 and an activation pattern from 24 h to 48 h, plot the time series of quality through node n352*"; and (3) **Change order**, "*simulate water quality after applying a chemical booster at n2 that maintains a strength of 1000 between 24 h to 48 h using the SETPOINT method, and plot the resulting quality time series at n352*". Each variant was executed five times. As summarised in Table 5, the system maintained a 100 percent task success rate and 100 percent tool invocation accuracy, with no human interventions required. The only variation occurred in the average number of code generation attempts, with CodeRunner typically completing tasks within one to two attempts. The Missing Sentence variation produced a slightly higher average of 1.4 attempts, suggesting that incomplete instructions introduced minor interpretive ambiguity while having no effect on task completion.

Additionally, LLMs are also susceptible to hallucination, generating plausible but incorrect or irrelevant outputs (Darwish et al., 2025). To examine this behaviour in EPANET-Agentic, task (b) from Section 4.4 was modified in two ways: (1) replacing the keyword "SETPOINT" with a natural language description, "*maintains a fixed concentration at node's outflow*", and (2) requiring the task to plot water quality distribution using the WNTR "*wntr.graphics.plot_network*" function. Under normal conditions, EPANET-Agentic solves this task by having TaskExecutor add the chemical booster scenario, followed by CodeRunner generating the plotting code. However, Fig. 8 highlights two representative hallucination cases observed during these tests. When the definition of "SETPOINT" was not explicitly provided in the TaskExecutor's system prompt, the agent incorrectly used the "CONCEN" method. Similarly, when CodeRunner lacked accurate knowledge of the wntr.graphics.plot_network function, it defaulted to using matplotlib, producing blank output images despite error-free execution. These results directly demonstrate that such behaviours arise when domain-specific terminology or functions are absent from the agent's prompt or context.

5. Discussion

5.1. Advantages of tool-driven nested design

Many multi-agent architectures hardwire which agents interact with which others, providing scalability and modularity but producing

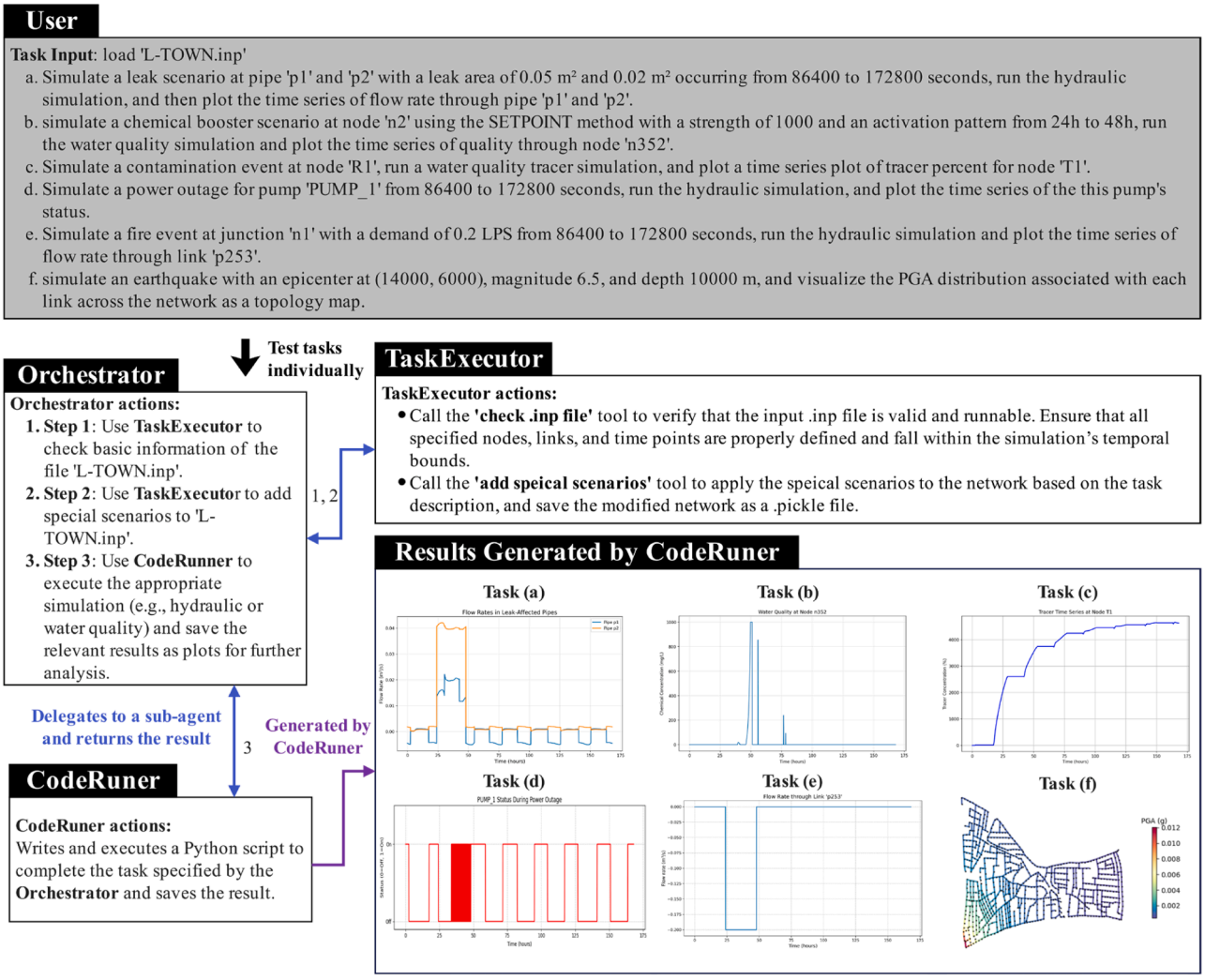


Fig. 7. Six illustrative Scenario Simulation tasks using EPANET-Agentic: (a) leak, (b) chemical booster, (c) tracer contamination, (d) pump power outage, (e) fire demand, and (f) earthquake.

Table 4
Trustworthiness verification of EPANET-Agentic across different test tasks.

Task Category (Number of Tasks)	System Characteristics (11)			System Dynamics (11)			System Operation (8)			Scenarios Simulations (9)		
WDN	L-Town	C-Town	Net3	L-Town	C-Town	Net3	L-Town	C-Town	Net3	L-Town	C-Town	Net3
Avg. Code Attempts	1.4	0.9	1	1.4	1.7	1.6	1.4	1.5	1.3	1.1	1.4	1.3

Note: Tool invocation accuracy (100%), human intervention (0), and success rate (100%) were consistent across all tasks and therefore omitted from the table for conciseness.

Table 5
Results of the Semantic Stress Test, evaluating the sensitivity of EPANET-Agentic to variations in task descriptions.

Type	Base	No punctuation	Missing sentence	Change order
Avg. Code Attempts	1.2	1.1	1.4	1.2

Note: Tool invocation accuracy (100%), human intervention (0), and success rate (100%) were consistent across all tasks and therefore omitted from the table for conciseness.

brittle, task specific pipelines with limited adaptability (Jimenez-Romero et al., 2025). By contrast, systems such as Microsoft's Magentic-One employ a central orchestrator that selects the next agent at runtime, improving flexibility; however, this choice is governed by policies conditioned on prompts, leaving behaviour susceptible to prompt drift, ambiguity, and limited control granularity (Fourney et al., 2024). Building on these insights, this study adopts the tool-driven nested agent design described in Section 2.2, in which the Orchestrator invokes sub-agents through predefined, typed function-call interfaces that present each agent as a callable tool with a unique name and concise description. This design offers several key advantages: (1) by decomposing complex tasks into structured tool calls with well-defined parameters, the design minimizes reliance on extended conversational context, mitigating issues, such as context overflow and hallucinations commonly in LLM-based reasoning (Darwish et al., 2025); (2) predefined key parameters combined with integrated validation mechanisms help ensure tool outputs align with user intent and prevent unintended behaviours (Ghafarollahi and Buehler, 2025),

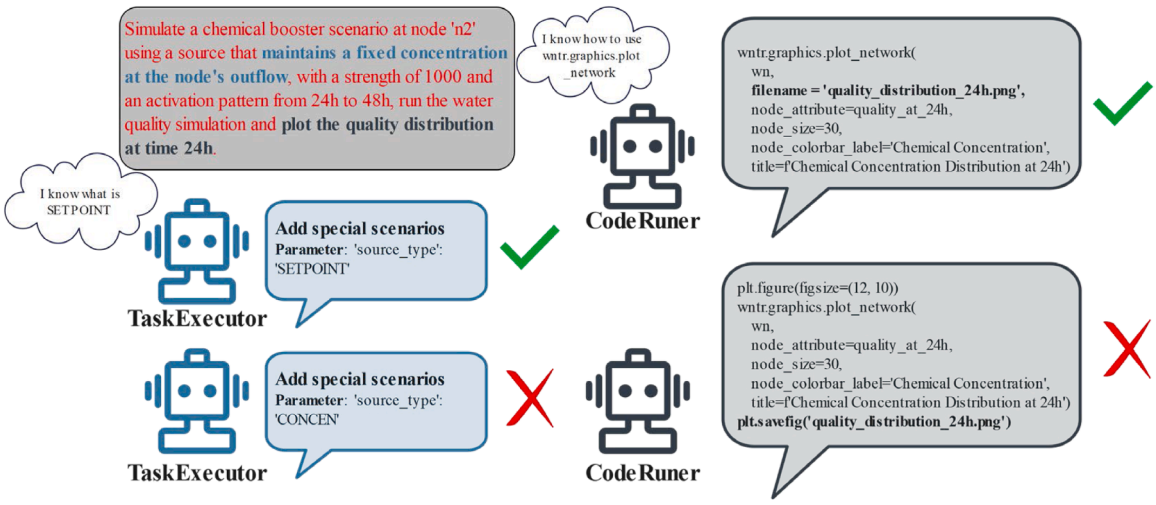


Fig. 8. Representative cases of LLM hallucinations observed in EPANET-Agentic when domain-specific instructions are incomplete.

enhancing reliability in engineering workflows; and (3) its modular, robust, and composable architecture supports seamless integration of new tools or agents without model retraining, facilitating adaptability and cross-domain scalability (Zhang et al., 2025). Collectively, these properties enable EPANET-Agentic to provide a safe, controllable, and scalable platform for orchestrating complex WDN simulations, aligning with best practices for multi-agent systems in critical domains.

5.2. *Human-in-the-Loop mechanisms and hallucination mitigation strategies*

Although the Semantic Stress Test in Section 4.6 shows that the system performs strongly, real-world usage involving diverse users may introduce greater semantic variability, increasing the risk of execution errors. Thus, integrating a human-in-the-loop mechanism into EPANET-Agentic is essential. In the current design of EPANET-Agentic, every tool or sub-agent invocation requires explicit user approval, which prevents unintended simulation operations and ensures that no physically invalid hydraulic computation can proceed without human verification. This mechanism allows continuous monitoring and intervention, such as clarifying ambiguous inputs, adjusting execution plans, or correcting outputs, ensuring reliability in safety-critical applications like water distribution systems. Although many studies (Feng et al., 2025; Goldshtein et al., 2025) focus on fully automated agent systems, the unpredictability of LLMs necessitates human oversight to ensure robustness and reliability in complex or safety-critical applications such as water distribution systems.

Additionally, the hallucination case in Section 4.6 underscores the importance of precise system prompts in constraining agent behaviour and mitigating hallucinations. However, due to context length limitations, it is impractical to embed all necessary domain knowledge directly into prompts (Wang et al., 2024). In this study, EPANET-Agentic further reduces hallucination risks by using a strict system message that explicitly forbids the LLM from fabricating numbers and by restricting the LLM from generating any numerical results. All hydraulic values must be produced by predefined Python functions implemented with the WNTR library, which ensures full compliance with EPANET's physical and modelling constraints and prevents the system from producing physically impossible outcomes. Beyond the safeguards adopted in this study, several complementary strategies have been explored in recent work. Goldshtein et al. (2025) incorporated Retrieval-Augmented Generation (RAG) into the LLM-EPANET architecture, effectively mitigating reasoning errors caused by knowledge gaps in water distribution system modelling and significantly improving model accuracy in handling complex queries. In contrast, Darwish et al. (2025) proposed a multi-agent framework that combines external API tool grounding with rule-based verification, utilising a consultant-evaluator agent structure to effectively reduce LLM hallucinations and enhance system stability in practical deployments. More recently, Microsoft's GraphRAG (Edge et al., 2025) has demonstrated how graph-based retrieval structures can further improve contextual reasoning by organising knowledge to capture relationships between entities, enabling agents to handle complex tasks with higher precision. Therefore, as EPANET-Agentic continues to expand, integrating advanced knowledge retrieval and context management techniques to enhance its reasoning accuracy and robustness will be an inevitable trend.

5.3. *Limitations and future directions*

This study integrates LLMs' advanced reasoning, planning, and multimodal capabilities with EPANET's simulation environment to develop EPANET-Agentic, a natural language-controlled multiple agent-based system for WDN analysis. Experimental results across multiple task categories demonstrate high success rates, accurate tool invocation, and minimal need for human intervention. These results indicate the potential to lower the operational barrier to complex hydraulic modelling by allowing natural-language control that reduces the need for EPANET-specific operational skills for model editing, control setup, and automated scenario runs, while maintaining engineering-grade reliability, with broader accessibility to non-expert users remaining a future direction. The tool-driven nested agent architecture reduces hallucination risks, enhances workflow transparency, and ensures modular scalability, thereby enabling seamless coordination between agents and tools to complete a wide range of WDN analytical tasks. However, as the current system represents an early-stage prototype, its functionality remains limited to routine EPANET operations rather than the full spectrum of modelling and decision-support tasks required by different water-sector stakeholders.

Beyond these findings, several limitations must be acknowledged. First, EPANET-Agentic currently processes only predefined .inp files and cannot autonomously construct network models from heterogeneous data. Second, performance remains sensitive to the design of human prompts, with susceptibility to subtle hallucinations when domain-specific instructions or tool descriptions are incomplete. Third, evaluation has been limited to benchmark networks, its performance on large-scale WDNs with thousands of elements remains unverified, where computational efficiency and memory usage may become critical constraints. Finally, the system operates offline without real-time data integration or online decision-making capabilities, both of which are essential for practical deployment in operational water utilities and its

integration as a digital twin. These limitations reflect the system's current focus on supporting operators and technicians with basic analytical tasks, while more advanced functionalities for researchers, planners, and decision-makers are yet to be developed.

To address these limitations, future research needs to proceed along several directions. First, the integration of SCADA and GIS systems, together with real-time sensor data streams and online learning mechanisms, is expected to enable EPANET-Agentic to automatically construct, update, and optimise EPANET input files while supporting continuous decision-making, thereby moving EPANET-Agentic closer to proactive network management. Second, incorporating RAG and graph-based knowledge representations will enhance contextual reasoning and reduce the system's reliance on static prompts, consequently mitigating further the risk of hallucinations (Aquino et al., 2025; Arslan et al., 2024; Edge et al., 2025). Third, replacing general-purpose LLMs with distilled, domain-specialized small language models may significantly improve computational efficiency and reliability in hydraulic engineering applications (Shen et al., 2025), while also addressing the practical requirement for on-premises deployment. Fourth, future versions of EPANET-Agentic may incorporate emerging research on evolving agent systems (Fang et al., 2025) that continually improve their performance through interaction with the environment, offering a path toward enhanced robustness as the system is used over time. In parallel, practical implementations could develop user feedback loops, visualisation tools for monitoring workflow execution, and adaptive mechanisms to handle semantic variability in natural-language prompts, helping to balance automation with necessary human oversight in safety-critical WDN applications. Finally, extending the multi-agent system to other simulation domains, such as sewer networks and urban drainage systems, will broaden its applicability. With additional domain modules, deeper integration of LLM reasoning abilities, and improved task diversification, EPANET-Agentic aims to gradually expand its target audience to include researchers, students, and decision-makers and to support a more comprehensive suite of scientific, operational, and planning workflows. With its modular architecture and inherent ability for continuous evolution, EPANET-Agentic holds great promise as a next-generation solution for autonomous, intelligent, and resilient WDNs management.

6. Conclusions

This study proposed EPANET-Agentic, a novel multi-agent system that combines the advanced reasoning and planning capabilities of LLMs with the robust simulation environment of EPANET. Results from a series of 39 test tasks, including System Characteristics, System Dynamics, System Operation, and Scenario Simulation, indicate that EPANET-Agentic delivers accurate, reproducible simulations under natural-language control. It achieved a 100 percent task success rate and 100 percent tool invocation accuracy across all scenarios, confirming that it retains EPANET's computational reliability while enabling natural-language operation. Supported by a tool-driven design that minimises hallucination risk and offers modular extensibility, the system is positioned for broader integration and applications, such as Digital Twin use in WDN management, thereby charting a practical path toward live, data-driven deployments.

Overall, EPANET-Agentic highlights the potential of LLM-powered multi-agent systems to streamline hydraulic modelling workflows by replacing software-specific operational steps with natural-language interaction, without compromising numerical correctness. Nevertheless, the current system remains limited by its dependence on predefined input files and offline operation. Future work will aim to integrate SCADA/GIS systems, real-time adaptive learning, and retrieval-augmented reasoning, and a dedicated safeguarding layer for continuous validation and risk management, which will further enhance its functionality and establish EPANET-Agentic as a next-generation solution for autonomous, intelligent, and resilient WDNs management.

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

During the preparation of this work, the author(s) used GPT 5 in order to improve language and readability. 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 publication.

Data availability

All source data, agent conversation logs (chat transcripts) used for evaluation, and the implementation code that support the findings of this study are available at https://github.com/wangjian169/EPANET-Agentic.

CRediT authorship contribution statement

Jian Wang: Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation. **Guangtao Fu:** Writing – review & editing, Supervision, Funding acquisition. **Dragan Savic:** Writing – review & editing, Supervision, Funding acquisition.

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.

Acknowledgements

Jian Wang is a PhD student sponsored by the China Scholarship Council (202408340010) at the University of Exeter. Dragan Savic has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. [951424]). Guangtao Fu has received funding from the British Council under the UK-China Institutional Partnership. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.watres.2026.125433.

References

Acharya, D.B., Kuppan, K., Divya, B., 2025. Agentic AI: autonomous intelligence for complex goals—a comprehensive survey. IEEE Access 13, 18912–18936. https://doi.org/10.1109/ACCESS.2025.3532853.

Aquino, G.D.A.E., Azevedo, N.D.S.D., Okimoto, L.Y.S., Camelo, L.Y.S., Bragança, H.L.D.S., Fernandes, R., Printes, A., Cardoso, F., Gomes, R., Torné, I.G., 2025. From RAG to Multi-Agent systems: a survey of modern approaches in LLM development. https://doi.org/10.20944/preprints202502.0406.v1.

Arslan, M., Ghanem, H., Munawar, S., Cruz, C., 2024. A survey on RAG with LLMs. Procedia Comput. Sci. 246, 3781–3790. https://doi.org/10.1016/j.procs.2024.09.178.

Bilal Pant, M., Snasel, V., 2021. Design optimization of water distribution networks through a novel differential evolution. IEEE Access. 9, 16133–16151. https://doi.org/10.1109/ACCESS.2021.3052032.

Darwish, A.M., Rashed, E.A., Khoriba, G., 2025. Mitigating LLM hallucinations using a multi-agent framework. Information 16, 517. https://doi.org/10.3390/info16070517.

DeepSeek-AI, 2024. DeepSeek-V3 technical report (Technical Report No. arXiv: 2412.19437). arXiv.

DeepSeek-AI, Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., Gao, H., Gao, K., Gao, W., Ge, R., Guan, K., Guo, D., Guo, J., Hao, G., Hao, Z., He, Y., Hu, W., Huang, P., Li, E., Li, G., Li, J., Li, Y., Li, Y.K., Liang, W., Lin, F., Liu, A.X., Liu, B., Liu, W., Liu, Xiaodong, Liu, Xin, Liu, Y., Lu, H., Lu, S., Luo, F., Ma, S., Nie, X., Pei, T., Piao, Y., Qiu, J., Qu, H., Ren, T., Ren, Z., Ruan, C., Sha, Z., Shao, Z., Song, J., Su, X., Sun, J., Sun, Y., Tang, M., Wang, B., Wang, P., Wang, S., Wang, Yaohui, Wang, Yongji, Wu, T., Wu, Y., Xie, X., Xie, Zhenda, Xie, Ziwei, Xiong,

Y., Xu, H., Xu, R.X., Xu, Y., Yang, D., You, Y., Yu, S., Yu, X., Zhang, B., Zhang, H., Zhang, Lecong, Zhang, Liyue, Zhang, Mingchuan, Zhang, Minghua, Zhang, W., Zhang, Y., Zhao, C., Zhao, Y., Zhou, Shangyan, Zhou, Shunfeng Zhu, Q., Zou, Y., 2024. DeepSeek LLM: scaling open-source language models with longtermism. https://doi.org/10.48550/arXiv.2401.02954.

DeepSeek-AI, Guo, D., Yang, D., Zhang, Haowei, Song, J., Zhang, Ruoyu, Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z.F., Gou, Z., Shao, Z., Li, Zhuoshu, Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, Jiashi, Wang, J., Chen, Jingchang, Yuan, J., Qiu, J., Li, Junlong, Cai, J.L., Ni, J., Liang, J., Chen, Jin, Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, Lean, Zhang, Lecong, Zhao, L., Wang, Litong, Zhang, Liyue, Xu, L., Xia, L., Zhang, Mingchuan, Zhang, Minghua, Tang, M., Li, Meng, Wang, M., Li, Mingming, Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, Ruisong, Pan, R., Wang, R., Chen, R.J., Jin, R.L., Chen, R., Lu, S., Zhou, Shangyan, Chen, S., Ye, S., Wang, S., Yu, S., Zhou, Shunfeng, Pan, S., Li, S.S., Zhou, Shuang, Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W.L., An, W., Liu, Xiaodong, Wang, Xiaohan, Chen, Xiaokang, Nie, X., Cheng, X., Liu, Xin, Xie, X., Liu, Xingchao, Yang, X., Li, X., Su, X., Lin, X., Li, X.Q., Jin, X., Shen, X., Chen, Xiaosha, Sun, X., Wang, Xiaoxiang, Song, X., Zhou, X., Wang, Xianzu, Shan, X., Li, Y.K., Wang, Y.Q., Wei, Y. X., Zhang, Yang, Xu, Y., Li, Yao, Zhao, Y., Sun, Y., Wang, Yaohui, Yu, Y., Zhang, Yichao, Shi, Y., Xiong, Yiliang, He, Ying, Piao, Y., Wang, Yisong, Tan, Y., Ma, Yiyang, Liu, Yiyuan, Guo, Y., Ou, Y., Wang, Yuduan, Gong, Y., Zou, Y., He, Yujia, Xiong, Yunfan, Luo, Y., You, Y., Liu, Yuxuan, Zhou, Y., Zhu, Y.X., Xu, Y., Huang, Y., Li, Yaohui, Zheng, Y., Zhu, Y., Ma, Yunxian, Tang, Y., Zha, Y., Yan, Y., Ren, Z.Z., Ren, Z., Sha, Z., Fu, Z., Xu, Zhean, Xie, Zhenda, Zhang, Zhengyan, Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Zilin, Xie, Ziwei, Song, Z., Pan, Z., Huang, Z., Xu, Zhipeng, Zhang, Zhongyu, Zhang, Zhen, 2025. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. https://doi.org/10.48550/arXiv.2501.12948.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R.O., Larson, J., 2025. From local to global: a graph RAG approach to query-focused summarization. https://doi.org/10.48550/arXiv.2404.16130.

Fang, J., Peng, Y., Zhang, X., Wang, Y., Yi, X., Zhang, G., Xu, Y., Wu, B., Liu, S., Li, Z., Ren, Z., Aletras, N., Wang, X., Zhou, H., Meng, Z., 2025. A comprehensive survey of self-evolving AI agents: a new paradigm bridging foundation models and lifelong agentic systems. https://doi.org/10.48550/arXiv.2508.07407.

Feng, J., Xu, R., Chu, X., 2025. OpenFOAMGPT 2.0: end-to-end, trustworthy automation for computational fluid dynamics. https://doi.org/10.48550/arXiv.2504.19338.

Fourney, A., Bansal, G., Mozannar, H., Tan, C., Salinas, E., Erkang, Zhu, Niedtner, F., Proebsting, G., Bassman, G., Gerrits, J., Alber, J., Chang, P., Loynd, R., West, R., Dibia, V., Awadallah, A., Kamar, E., Hosn, R., Amershi, S., 2024. Magentic-one: a generalist multi-agent system for solving complex tasks. https://doi.org/10.48550/arXiv.2411.04468.

Fu, G., 2025. Towards autonomous planning and management of urban water systems. https://doi.org/10.22541/essoar.175745445.50927919/v1.

Fu, G., Savic, D., Butler, D., 2024. Making waves: towards data-centric water engineering. Water Res. 256, 121585. https://doi.org/10.1016/j.watres.2024.121585.

Ghafarollahi, A., Buehler, M.J., 2025. Automating alloy design and discovery with physics-aware multimodal multiagent AI. Proc. Natl. Acad. Sci. U.S.A. 122, e2414074122. https://doi.org/10.1073/pnas.2414074122.

Goldshtein, Y., Perelman, G., Schuster, A., Ostfeld, A., 2025. Large language models for water distribution systems modeling and decision-making. https://doi.org/10.48550/arXiv.2503.16191.

Hedaiaty Marzouny, N., Dziedzic, R., 2024. AI-assisted pump operation for energy-efficient water distribution systems. In: The 3rd International Joint Conference on Water Distribution Systems Analysis & Computing and Control for the Water Industry (WDSA/CCWI 2024). Presented at the International Joint Conference on Water Distribution Systems Analysis & Computing and Control for the Water Industry. MDPI, p. 3. https://doi.org/10.3390/engproc2024069003.

Jimenez-Romero, C., Yegenoglu, A., Blum, C., 2025. Multi-agent systems powered by large language models: applications in swarm intelligence. https://doi.org/10.48550/arXiv.2503.03800.

Khedr, A., Tolson, B., 2016. Comparing optimization techniques with an engineering judgment approach to WDN design. J. Water Resourc. Plann. Manage. 142, C4015014. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000611.

Klise, K., Hart, D., Bynum, M., Hogge, J., Haxton, T., Murray, R., Burkhardt, J., 2020. Water network tool for resilience (WNTR). User Manual, Version 0.2.3. p. SAND–2020-9301R, EPA/600/R–20/185, 1660790. https://doi.org/10.2172/1660790.

Klise, K.A., Bynum, M., Moriarty, D., Murray, R., 2017. A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study. Environment. Modell. Software. 95, 420–431. https://doi.org/10.1016/j.envsoft.2017.06.022.

Lyu, H., Zhou, S., Wang, Z., Fu, G., Zhang, C., 2025. Assessing large multimodal models for urban floodwater depth estimation. https://doi.org/10.22541/essoar.173315699.90284357/v1.

Marques, J., Cunha, M., Savić, D., 2018. Many-objective optimization model for the flexible design of water distribution networks. J. Environ. Manage. 226, 308–319. https://doi.org/10.1016/j.jenvman.2018.08.054.

Monsef, H., Naghashzadegan, M., Jamali, A., Farmani, R., 2019. Comparison of evolutionary multi objective optimization algorithms in optimum design of water distribution network. Ain Shams Eng. J. 10, 103–111. https://doi.org/10.1016/j.asej.2018.04.003.

OpenAI, 2025. GPT-5 system card (Technical report). OpenAI, San Francisco, CA.

Rossman, L., Woo, H., Tryby, M., Shang, F., Janke, R., Haxton, T., 2020. EPANET 2.2 User Manual. U.S. Environmental Protection Agency, Washington, DC.

Safitri, A., Wahyudi, S.I., Soedarsono, 2023. Simulation of pipe networks using EPANET to optimize water supply: a case study for Arjawinangun area. Indonesia. Archive. Hydro-Eng. Environment. Mechanic. 70, 17–28. https://doi.org/10.2478/heem-2023-0002.

Sela, L., Sowby, R.B., Salomons, E., Housh, M., 2025. Making waves: the potential of generative AI in water utility operations. Water Res. 272, 122935. https://doi.org/10.1016/j.watres.2024.122935.

Seo, M., Baek, J., Lee, S., Hwang, S.J., 2025. Paper2Code: automating code generation from scientific Papers in Machine Learning. https://doi.org/10.48550/arXiv.2504.17192.

Shen, L., Yang, Q., Huang, X., Ma, Z., Zheng, Y., 2025. GPIoT: tailoring small language models for IoT program synthesis and development. In: Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems. Presented at the SenSys '25: 23rd ACM Conference on Embedded Networked Sensor Systems. ACM, UC Irvine Student Center, Irvine CA USA, pp. 199–212. https://doi.org/10.1145/3715014.3722064.

Sirsant, S., Reddy, M.J., 2020. Assessing the performance of surrogate measures for water distribution network reliability. J. Water Resourc. Plan. Manage. 146, 04020048. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001244.

Taormina, R., van der Werf, J.A., 2024. Interpretable sewer defect detection with large multimodal models. Eng. Proceed. 69, 158. https://doi.org/10.3390/engproc2024069158.

Tsiami, L., Makropoulos, C., Savic, D., 2025. Rethinking Urban Water Network Design: a reinforcement learning framework for long-term flexible planning. Water Resour. Manage. https://doi.org/10.1007/s11269-025-04290-8.

Vrachimis, S.G., Eliades, D.G., Taormina, R., Kapelan, Z., Ostfeld, A., Liu, S., Kyriakou, M., Pavlou, P., Qiu, M., Polycarpou, M.M., 2022. Battle of the leakage detection and isolation methods. J. Water Resource. Plann. Manag. 148, 04022068. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001601.

Wang, J., Fu, G., Savic, D., 2026. Leveraging large language models for automating water distribution network optimization. Water Res. 288, 124536. https://doi.org/10.1016/j.watres.2025.124536.

Wang, J., Liu, L., Savic, D., Fu, G., 2025. Heterogeneous graph neural networks enhance pressure estimation in water distribution networks. Water Res., 123843 https://doi.org/10.1016/j.watres.2025.123843.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W.X., Wei, Z., Wen, J.-R., 2024. A Survey Large Lang. Model Based Autonomous Agents. Front. Comput. Sci. 18, 186345. https://doi.org/10.1007/s11704-024-40231-1.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Qin, W., Zheng, Y., Qiu, X., Huang, X., Zhang, Q., Gui, T., 2025. The rise and potential of large language model based agents: a survey. Sci. China Inf. Sci. 68, 121101. https://doi.org/10.1007/s11432-024-4222-0.

University of Exeter, 2026. Benchmarks. https://www.exeter.ac.uk/research/centres/cws/resources/benchmarks/ (accessed: 21-01-2026).

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, Jian, Tu, J., Zhang, J., Yang, Jianxin, Yang, Jiaxi, Zhou, Jing, Zhou, Jingren, Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, Mei, Xue, M., Li, Mingze, Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, Xingzhang, Wang, X., Zhang, X., Ren, Xuancheng, Fan, Y., Su, Y., Zhang, Yichang, Zhang, Yinger, Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z., 2025. Qwen3 Technical report. https://doi.org/10.48550/arXiv.2505.09388.

Zhang, W., Cui, C., Zhao, Y., Hu, R., Liu, Y., Zhou, Y., An, B., 2025. AgentOrchestra: a hierarchical multi-agent framework for general-purpose task solving. https://doi.org/10.48550/arXiv.2506.12508.