AeroNoxis: Retrieval-Grounded Multi-Agent Design of Industrial Waste-Gas Treatment Trains

Anonymous Author(s)

Abstract

Industrial waste-gas treatment train design requires the integration of pollutant profiles, source-specific operating conditions, engineering constraints, and dispersed technical evidence. Existing large language models and generic retrieval-augmented generation systems can support environmental question answering, but they are not designed to generate auditable, constraint-aware treatment train configurations. In this paper, we propose AeroNoxis, a retrieval-grounded multi-agent framework for early-stage industrial waste-gas treatment train design. AeroNoxis decomposes the design process into scenario parsing, hierarchical evidence retrieval, candidate train planning, critic-based constraint verification, and evidence-bound explanation. Given an emission scenario, the framework generates ranked candidate treatment trains, maps treatment units to target pollutants, checks process-level constraints, and reports assumptions, risks, and uncertainty. To evaluate this task, we construct APCTrainBench, a benchmark for industrial air-pollution-control train design across scenarios of varying complexity. Experiments show that AeroNoxis consistently improves treatment train quality over stand-alone language models and single-agent retrieval baselines, with especially large gains in difficult cases requiring process ordering and constraint reasoning. Ablation results further demonstrate that the hierarchical knowledge base and critic agent play complementary roles in improving evidence grounding, train completeness, and engineering feasibility. These results suggest that retrieval-grounded multi-agent reasoning can provide transparent and auditable decision support for early-stage industrial air-pollution-control design.

CCS Concepts

• **Applied computing** → **Environmental sciences**; • **Computing methodologies** → *Artificial intelligence*; • **Information systems** → *Decision support systems*.

Keywords

Multi-agent reasoning, Industrial waste-gas treatment, Constrained decision support, RAG

ACM Reference Format:

Anonymous Author(s). 2026. AeroNoxis: Retrieval-Grounded Multi-Agent Design of Industrial Waste-Gas Treatment Trains. In *Proceedings of KDD 2026 Workshop SciSoc Agents & LLMs (KDD '26).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '26, Jeju, Korea

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2026/08
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Industrial waste-gas treatment train design is a complex early-stage engineering task that requires aligning emission characteristics, pollutant-control objectives, operating conditions, and practical engineering constraints [1–3]. Unlike single-unit technology selection, treatment train design must determine how multiple control units should be combined, ordered, and justified under uncertain or incomplete site information [4, 5]. In practice, this process still relies heavily on expert experience and dispersed technical documents, making it difficult to consistently synthesize evidence across source conditions, regulatory requirements, operational feasibility, safety risks, and secondary pollution concerns [6–8]. As a result, preliminary designs may overlook key pollutant-unit matches, violate process constraints, or provide recommendations that are difficult to audit [9, 10]. These challenges motivate the need for an evidence-grounded and constraint-aware decision-support framework that can organize heterogeneous engineering knowledge and generate transparent candidate treatment trains for early-stage screening.

Recent advances in large language models provide a potential pathway for assisting this type of knowledge-intensive engineering reasoning [11, 12]. LLMs can interpret natural-language descriptions, summarize technical evidence, and generate preliminary design suggestions from heterogeneous sources [13, 14]. Retrieval-augmented generation further improves this process by linking model outputs to external documents, which is especially valuable when design decisions depend on technical specifications, regulatory context, and prior engineering cases [15–17]. However, generic LLM or RAG systems are not inherently designed for treatment train design. They may retrieve relevant facts but fail to organize them into a valid process sequence, distinguish hard engineering constraints from soft preferences, or verify whether each target pollutant is covered by an appropriate control unit [18]. They also often provide plausible recommendations without explicit evidence binding, uncertainty reporting, or systematic checks for safety and secondary pollution risks [19]. Therefore, simply applying LLM-based question answering to industrial waste-gas control is insufficient for producing transparent, constraint-aware, and auditable early-stage design support.

In this paper, we introduce AeroNoxis, a retrieval-grounded multi-agent framework for early-stage industrial waste-gas treatment train design. Rather than treating the task as a direct question-answering problem, AeroNoxis formulates it as a constrained design process in which each recommendation must be supported by retrievable engineering evidence and tested against process-level feasibility. Given an emission scenario, the framework interprets the source conditions and control objectives, retrieves evidence at different levels of engineering abstraction, and uses this evidence to reason from pollutant characteristics to feasible treatment configurations. The retrieved knowledge is therefore not used merely as loose textual context for generation, but as a basis for examining how control units can be combined under practical operating

constraints. This enables AeroNoxis to generate candidate treatment trains whose assumptions, limitations, and potential risks can be inspected, rather than producing a single plausible but weakly justified answer. To evaluate the framework, we construct APC-TrainBench, a benchmark for industrial air-pollution-control train design under scenarios with varying complexity and incomplete information. Through comparisons with stand-alone LLMs, single-agent RAG systems, and ablated multi-agent variants, we examine whether retrieval-grounded agentic reasoning can improve the technical coherence, evidence support, robustness, and auditability of early-stage waste-gas treatment design.

2 Related works

LLMs for environmental decision support and engineering explanation: Large language models have recently been explored as tools for environmental decision support, especially in tasks that require integrating technical knowledge, environmental models, and human decision-making [11–14]. Nie and Liu proposed LLM-assisted and LLM-driven frameworks for environmental decision-making and showed, through a PFAS-related water management case, that LLMs can support human experts but still face limitations when complex optimization and domain constraints must be handled automatically [20]. In the atmospheric domain, AirGPT integrates conversational AI with a curated atmospheric-science corpus and specialized analytical tools to support air-quality assessment, demonstrating higher accuracy than general-purpose LLMs in domain-specific queries [21]. Related systems have also used RAG, function calling, and domain adaptation to manage atmospheric-emission knowledge and interpret pollution-control guidelines, indicating that retrieval grounding can reduce hallucination and improve technical relevance [22]. These studies suggest that LLMs can make environmental knowledge more accessible and explainable.

Multi-agent systems for atmospheric and air-quality applications: Multi-agent systems have been widely explored for air-quality monitoring, pollution simulation, policy evaluation, and adaptive sensing. Earlier agent-based air-pollution simulators model emission sources, monitoring agencies, and control actors as interacting agents, often coupled with dispersion or prediction modules to compare emission-control strategies under different scenarios [23–25]. More recent work extends this direction to tool-augmented and LLM-enhanced monitoring, where external data sources and analytical functions are incorporated into conversational air-quality interfaces [18, 26]. Cooperative multi-agent reinforcement learning has also been used to coordinate autonomous drones for air-pollution mapping, demonstrating the value of agent collaboration in dynamic environmental sensing tasks [27].

Beyond air pollution, RAG-based multi-agent LLM systems have been developed for natural-hazard resilience and wildfire-risk analysis, showing that retrieval-grounded agents can support stakeholder-specific environmental decision support [28, 29]. However, these systems mainly focus on monitoring, simulation, forecasting, mapping, or risk communication. They do not address the synthesis of industrial waste-gas treatment trains, where agents must jointly retrieve engineering evidence, generate candidate process sequences, check hard constraints, rank alternatives, and explain risks. AeroNoxis addresses this gap by applying retrieval-grounded multi-agent reasoning to evidence-bound and constraint-aware early-stage treatment train design.

3 Methodology

3.1 The framework of AeroNoxis

AeroNoxis is a multi-agent retrieval-augmented framework with critic-based refinement for the design of industrial waste-gas treatment trains. First, user queries arrive in heterogeneous formats (free text, partial specifications, or mixed mode input), requiring formalization into a canonical representation before reasoning can proceed. Second, the design space grows combinatorially with the number of candidate treatment units, necessitating constrained generation. Third, single-pass generation does not provide an independent mechanism for detecting engineering violations, which requires explicit verification.

To address these challenges, AeroNoxis decomposes industrial waste-gas treatment train design into five stages handled by specialized agents (Task Parser, Retrieval, Planning, Critic, and Explanation). The agents operate sequentially but communicate through explicit intermediate representations, enabling iterative correction. Formally, let q denote the raw user input. AeroNoxis maps q to a canonical query representation $z = \mathcal{P}(q)$, defined as:

$$z = \langle \mathbf{p},\ \mathbf{w},\ \mathbf{g},\ \mathbf{k},\ \mathbf{m},\ \mathbf{a} \rangle, \tag{1}$$

where $\mathbf{p}$ denotes normalized contaminant identifiers, $\mathbf{w}$ flue-gas descriptors, $\mathbf{g}$ treatment targets, $\mathbf{k}$ operational and regulatory constraints, $\mathbf{m}$ missing field indicators, and $\mathbf{a}$ recorded assumptions. Conditioned on z, the Retrieval Agent returns a hierarchical evidence bundle,

$$B = \mathcal{R}(z) = \{B_u, B_t\}, \tag{2}$$

where B_u and B_t denote the unit-level and template-level evidence bundles. The Planning Agent generates a candidate set $C = \{c_1, \ldots, c_m\}$ conditioned on z and B. The Critic Agent evaluates each candidate and returns a constraint report. Candidates that pass or are repaired are forwarded to the Explanation Agent. If all candidates violate hard constraints, replanning is triggered for at most two rounds.

Algorithm 1 summarizes the end-to-end pipeline. Given a user query q, the Task Parser Agent maps heterogeneous input into the canonical query representation z. The Retrieval Agent then constructs a hierarchical evidence bundle $B = \{B_u, B_t\}$ by querying the unit-level and template-level indexes independently. Conditioned on z and B, the Planning Agent generates a diverse candidate set of industrial waste-gas treatment trains. Each candidate is subsequently examined by the Critic Agent, which returns a structured validation report and, when violations are reparable, revision signals for replanning. For surviving candidates, the Explanation Agent performs evidence binding, uncertainty estimation, and explanation generation. Finally, all candidates are ranked by contaminant coverage, constraint satisfaction, evidence support, and risk penalty to produce the final Top-K design report.

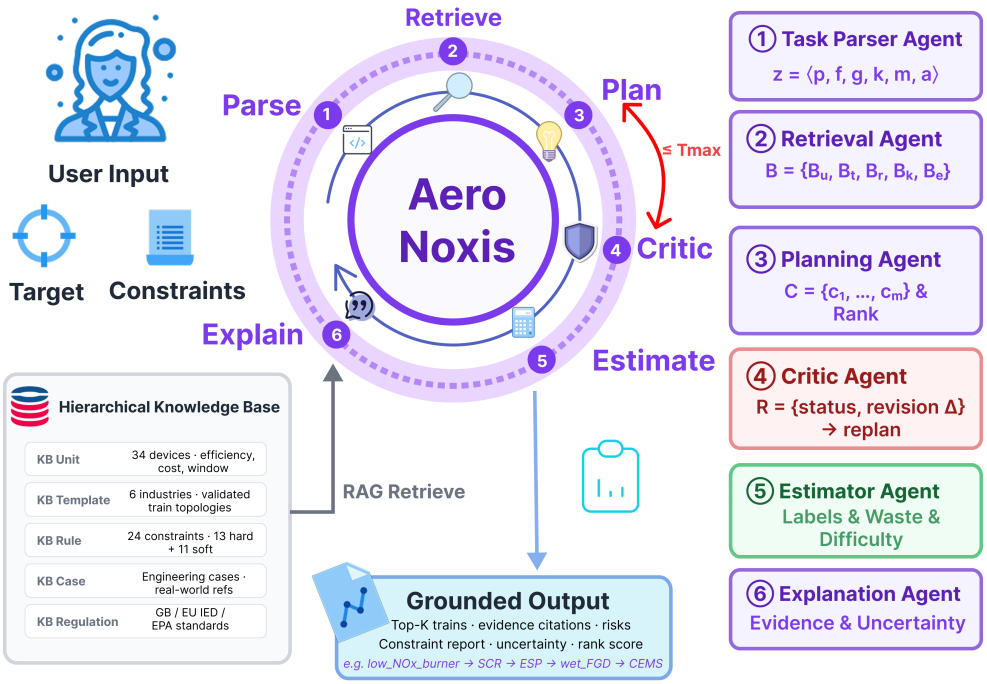

Figure 1: The AeroNoxis framework. The pipeline consists of six agents with iterative refinement: (1) Task Parser normalizes heterogeneous inputs, (2) Retrieval Agent constructs hierarchical evidence bundles from unit-level and template-level KB, (3) Planning Agent generates diverse candidates, (4) Critic Agent validates constraints, (5) Estimator Agent assigns cost-energy-operability labels, and (6) Explanation Agent produces ranked recommendations with evidence binding.

Algorithm 1 AeroNoxis: Retrieval-Grounded Multi-Agent Design of Industrial Waste-Gas Treatment Trains

Require: User query q
Ensure: Ranked Top-K treatment train report

$z \leftarrow \mathcal{P}(q)$ ▹ Task Parser: normalize input
$B \leftarrow \mathcal{R}(z) = \{B_u, B_t\}$ ▹ Retrieval: build evidence bundle
$\mathcal{C} \leftarrow \text{PlanningAgent}(z, B)$ ▹ Generate candidates
$T_{\text{iter}} \leftarrow 0$, $T_{\max} \leftarrow 2$
while $T_{\text{iter}} < T_{\max}$ **do**
  $\mathcal{V} \leftarrow \text{CriticAgent}(\mathcal{C}, z, B)$ ▹ Validate constraints
  if $\exists c \in \mathcal{C}$ with status $\in \{\text{PASS}, \text{WARN}\}$ **then**
    break ▹ At least one candidate survives
  else
    $\mathcal{C} \leftarrow \text{PlanningAgent}(z, B, \mathcal{V})$ ▹ Replan with revision signals
    $T_{\text{iter}} \leftarrow T_{\text{iter}} + 1$
  end if
end while
$\mathcal{C}_{\text{valid}} \leftarrow \{c \in \mathcal{C} : \text{status}(c) \in \{\text{PASS}, \text{WARN}\}\}$
for each $c \in \mathcal{C}_{\text{valid}}$ **do**
  $\text{EstimatorAgent}(c, z, B)$ ▹ Estimate cost, energy, operability
end for
$\mathcal{C}_{\text{ranked}} \leftarrow \text{RankCandidates}(\mathcal{C}_{\text{valid}}, z, B)$
$\text{Report} \leftarrow \text{ExplanationAgent}(\mathcal{C}_{\text{ranked}}, B)$ ▹ Evidence binding
return Top-K from Report

Task Parser Agent: User queries may arrive as free text, structured fields, or both, with inconsistent terminology. To ensure all downstream agents operate on a consistent factual basis, this agent maps any input into the canonical representation z. When free text is provided, an LLM-based prompt extracts a structured JSON object conforming to the query schema; when structured fields are available, they are consumed directly. Explicitly provided fields take precedence over inferred content. After extraction, contaminant names and aliases are mapped to canonical identifiers via a curated taxonomy, and measurement units are standardized. The parser populates the missing field set **m** and assumption set **a**, ensuring that ambiguities are explicitly recorded rather than silently propagated.

Retrieval Agent: To ground candidate generation in verifiable domain knowledge rather than implicit model memory, this agent constructs a preserving evidence bundle from the HKB. Retrieval is performed independently over the unit-level and template-level indexes, since unit-level evidence (process removal rates, operating ranges) and template-level evidence (validated train topologies) serve distinct roles in the reasoning pipeline. The case-level database is excluded to preserve benchmark independence. Retrieval follows a hybrid scoring scheme combining BM25 matching [30] with token-overlap similarity:

$$S_{\text{ret}}(z, d) = \alpha S^{\text{norm}}_{\text{BM25}}(z, d) + (1 - \alpha) S_{\text{overlap}}(z, d), \tag{3}$$

where d denotes a document chunk and $\alpha = 0.4$. Contaminant synonyms from the taxonomy are injected into the query term set to improve recall. Each retrieved chunk preserves its source identifier, relevance score, and coverage tags, enabling downstream agents to distinguish evidence-supported judgments from model inference.

Planning Agent: Owing to the number of possible ordered unit configurations grows exponentially, unconstrained generation is intractable. The Planning Agent generates multiple candidate treatment trains within a constrained design space defined by the controlled unit vocabulary and template-level priors from the HKB. Multiple candidates are produced because feasible treatment trains are rarely unique in practice. The planner (i) assembles candidates by composing unit-level evidence with template-level topological priors, (ii) enforces structural plausibility by maintaining logical ordering and chemical compatibility (e.g., coagulation before sedimentation), (iii) assigns functional roles linking units to contaminant removal objectives, and (iv) exposes candidate-level metadata (key units, brine generation, disinfection needs, energy labels) for downstream verification. LLM-based planning is attempted first; if valid candidates are insufficient, a template-guided heuristic constructor is used as fallback. Before constraint validation, the planner performs lightweight precorrection, such as removing brine producing units when $\mathbf{k}$ indicates unavailable brine disposal. Duplicates are removed before forwarding to the Critic Agent.

Critic Agent: To decouple generation from verification, this agent operates as a rule-based constraint verifier that evaluates each candidate against z, B, and codified engineering logic from the rule-level KB. Validation follows a two-tier procedure. Hard constraints enforce necessary conditions (exclusive use of units within the controlled taxonomy, presence of mandatory treatment functions for all target pollutants, safety-critical controls, process compatibility) and violations are grounds for rejection. Soft constraints capture technically feasible but penalized configurations, such as high energy trains in scenarios limited by energy in $\mathbf{k}$, high operational complexity, secondary waste generation, ammonia slip risk, or catalyst poisoning risk. When a hard constraint violation is reparable (e.g., a missing disinfection step can be added), the agent applies a targeted correction and assigns `WARN` status. Candidates still failing after repair are assigned `FAIL`. If all candidates fail, the Planning Agent is reinvoked with revision signals extracted from the failure report for at most two correction rounds ($T_{\max} = 2$). The output is a constraint report with candidate-level labels (`PASS`, `WARN`, `FAIL`) and revision directives.

Estimator Agent: To support early-stage screening, this agent assigns qualitative or semi-quantitative labels for cost, energy demand, operational complexity, and secondary waste generation to each validated candidate. The agent does not perform detailed engineering calculations or equipment sizing, but rather provides comparative indicators to help users prioritize candidates for further analysis. For each candidate c, the agent computes:

$$\text{Labels}(c) = \langle L_{\text{cost}}, L_{\text{energy}}, L_{\text{operability}}, W_{\text{secondary}} \rangle, \tag{4}$$

where $L_{\text{cost}} \in \{\text{low, medium, high, very high, unknown}\}$, $L_{\text{energy}} \in \{\text{low, medium, high, unknown}\}$, $L_{\text{operability}} \in \{\text{simple, moderate, complex}\}$ and $W_{\text{secondary}}$ is a list of secondary waste streams (e.g., FGD wastewater, spent catalyst, fly ash). Labels are derived from unit-level cost tags in the HKB, template-level precedents, and scenario constraints. When quantitative flow rate or concentration data are missing, the agent assigns `unknown` and records this limitation in the uncertainty field. The estimator explicitly does not replace detailed techno-economic analysis, material balance calculations, or vendor quotations. Its purpose is to provide relative comparison signals for candidate ranking, not absolute cost predictions. This limitation is clearly communicated in the final report to prevent misuse of preliminary estimates.

Explanation Agent: To ensure algorithmic transparency, this agent transforms validated candidates into feasible designs by binding each judgment to retrieved evidence with explicit provenance. It performs three functions. First, for each treatment train, it selects retrieved chunks whose coverage tags overlap with query contaminants and labels items as `evidence_backed` or `system_inference` by relevance score threshold. Second, it computes a ranking score for each candidate,

$$S_{\text{total}}(c) = 0.35\, S_{\text{cov}}(c) + 0.30\, S_{\text{con}}(c) + 0.25\, S_{\text{evidence}}(c) - P_{\text{risk}}(c), \tag{5}$$

where S_{cov} measures contaminant coverage, S_{con} reflects constraint satisfaction, S_{evidence} measures evidence support, and P_{risk} penalizes brine generation, high energy demand, or operational complexity. Third, it generates natural-language explanations (`why_it_works`, `risks`, `assumptions`) via an LLM prompt or template-based fallback. Uncertainty is estimated by evidence sufficiency. Designs with at least three evidence-based citations receive `LOW` uncertainty, one or two citations `MEDIUM`, and none `INSUFFICIENT_EVIDENCE`. The final output is a ranked Top-K report.

3.2 Cross-agent verification and iterative correction

AeroNoxis separates generation and verification across agents with distinct objectives, ensuring that no candidate enters the final report without independent scrutiny. The Planning Agent optimizes for diverse, evidence-informed generation; the Critic Agent evaluates constraint compliance and feasibility; the Explanation Agent verifies evidence grounding. When the Critic Agent detects reparable violations, it returns revision signals to the Planning Agent rather than terminating the pipeline. When all candidates are rejected, replanning is triggered using z with updated constraint directives from the failure report. The workflow alternates between generation and verification until an acceptable candidate survives or the maximum correction rounds ($T_{\max}$) are reached. This generate-verify-revise loop provides stronger robustness against unsupported designs than single-pass generation, while maintaining full auditability from input to recommendation.

3.3 Evaluation metrics

The evaluation unit is the final treatment-train recommendation produced by the agent, including pollutant-unit matching, treatment-unit coverage, sequence validity, and the quality of the ranked candidate list. We do not separately evaluate intermediate retrieval modules or general question-answering ability.

Table 1: Performance comparison of AeroNoxis and baselines. Numbers in parentheses are standard errors for five independent runs.

Method	P	R	F1	Hit Rate	NSS	ΔF1	ΔHit Rate	ΔNSS
GPT-5.3 [31]	$70.83_{(1.12)}$	$61.76_{(0.93)}$	$65.98_{(0.72)}$	$75.42_{(1.36)}$	$37.96_{(1.71)}$	–	–	–
Gemini-3.0-Pro [32]	$66.21_{(1.08)}$	$54.84_{(1.62)}$	$59.99_{(1.07)}$	$74.88_{(1.42)}$	$34.31_{(1.64)}$	–	–	–
Qwen3.5-Plus [33]	$63.56_{(1.17)}$	$53.32_{(0.83)}$	$57.99_{(0.69)}$	$74.17_{(1.51)}$	$35.72_{(1.79)}$	–	–	–
DeepSeek-V4 [34]	$64.03_{(1.21)}$	$54.18_{(1.05)}$	$58.69_{(0.80)}$	$73.52_{(1.48)}$	$35.08_{(1.83)}$	–	–	–
GLM-5 [35]	$68.44_{(1.25)}$	$60.22_{(1.31)}$	$64.07_{(0.92)}$	$73.91_{(1.56)}$	$36.36_{(1.88)}$	–	–	–
AeroNoxis-GPT-5.3	$\mathbf{82.31}_{(0.73)}$	$\mathbf{70.46}_{(0.81)}$	$\mathbf{75.93}_{(0.56)}$	$88.22_{(0.91)}$	$\mathbf{73.12}_{(1.18)}$	9.95	13.22	35.16
AeroNoxis-Gemini-3.0-Pro	$81.97_{(0.78)}$	$68.29_{(0.76)}$	$74.51_{(0.56)}$	$\mathbf{88.64}_{(0.95)}$	$68.63_{(1.21)}$	14.52	13.34	34.32
AeroNoxis-Qwen3.5-Plus	$81.70_{(0.81)}$	$66.55_{(1.04)}$	$73.35_{(0.71)}$	$87.96_{(0.98)}$	$66.31_{(1.24)}$	15.36	13.79	30.59
AeroNoxis-DeepSeek-V4	$81.36_{(0.84)}$	$66.27_{(0.95)}$	$73.04_{(0.67)}$	$87.59_{(1.02)}$	$71.88_{(1.29)}$	14.35	14.07	36.80
AeroNoxis-GLM-5	$80.94_{(0.89)}$	$67.64_{(1.13)}$	$73.69_{(0.77)}$	$88.14_{(1.06)}$	$72.37_{(1.34)}$	9.62	14.23	36.01

For each test case i, let G_i denote the set of reference design elements and $\hat{G}_i$ denote the set of design elements generated by the agent. Precision (P) measures how many generated elements are correct, while recall (R) measures how many reference elements are recovered:

$$P_i = \frac{|\hat{G}_i \cap G_i|}{|\hat{G}_i|}, \qquad R_i = \frac{|\hat{G}_i \cap G_i|}{|G_i|}. \tag{6}$$

The F1-Score is the harmonic mean of precision and recall:

$$F1_i = \frac{2P_iR_i}{P_i + R_i}. \tag{7}$$

HitRate evaluates whether the ranked agent output contains at least one acceptable design within the Top-K candidates:

$$\text{HitRate@}K = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\exists c \in \hat{C}_i^K,\ c \in C_i^*\right], \tag{8}$$

where $\hat{C}_i^K$ is the Top-K candidate set returned by the agent and C_i^* is the reference set of acceptable designs.

NSS measures normalized sequence similarity between the generated treatment train and the reference treatment train. For a predicted sequence $\hat{s}_i$ and a reference sequence s_i^*, we compute:

$$\text{NSS}_i = 1 - \frac{\text{EditDist}(\hat{s}_i, s_i^*)}{\max(|\hat{s}_i|, |s_i^*|)}. \tag{9}$$

A higher NSS indicates that the agent not only selects appropriate treatment units, but also places them in a more suitable engineering order. The final reported scores are averaged over all N test cases.

4 Experiments

We demonstrate the effectiveness of the waste-gas treatment train design, and these results prove the superior performance of AeroNoxis.

4.1 Datasets

We introduce **APCTrainBench**, a benchmark for industrial air-pollution-control treatment train design. The benchmark is designed to evaluate whether an LLM-based system can move beyond environmental question answering and perform structured engineering decision-making. Each instance describes an industrial waste-gas scenario, including the emission source, target pollutants, flue-gas conditions, design constraints, and expected output format. Given such an input, the model is required to generate ranked candidate treatment trains, together with pollutant-unit mappings, constraint analysis, risk descriptions, assumptions, and supporting evidence.

APCTrainBench contains 360 test cases in total, with 120 cases for each difficulty level: Easy, Middle, and Difficult. Easy cases usually involve a single dominant pollutant or a simple pollutant combination with standard control choices. Middle cases introduce multiple pollutants or one major engineering constraint, requiring the model to reason over both pollutant coverage and process feasibility. Difficult cases involve more complex industrial scenarios with multiple pollutants, incomplete operating conditions, safety concerns, secondary pollution risks, or strict design constraints. These cases are intended to test whether a system can produce technically plausible and auditable treatment trains under realistic uncertainty.

Compared with general environmental QA datasets, APCTrainBench emphasizes process-level reasoning rather than factual recall. A valid answer must not only mention relevant control technologies, but also place them in a reasonable sequence, explain which pollutants are addressed by each unit, identify potential engineering conflicts, and justify the recommendation with evidence. This makes APCTrainBench suitable for evaluating retrieval-grounded and agentic LLM systems that aim to support early-stage environmental engineering design.

4.2 Comparison with baselines

Table 1 compares AeroNoxis with five strong language-model baselines on APCTrainBench. Among the standalone models, GPT-5.3 achieves the strongest overall performance, with an F1 score of 65.98% and a hit rate of 75.42%. However, all standalone models show relatively low NSS scores, indicating that direct LLM generation can identify some relevant control units but still struggles to organize them into coherent treatment train sequences.

AeroNoxis consistently improves all backbone models across the main evaluation metrics. With GPT-5.3 as the backbone, AeroNoxis increases F1 from 65.98% to 75.93%, Hit Rate from 75.42% to 88.22%, and NSS from 37.96% to 73.12%. Similar gains are observed for

Gemini-3.0-Pro, Qwen3.5-Plus, DeepSeek-V4, and GLM-5. Across the five backbones, AeroNoxis improves F1 by 9.62% to 15.36% points, Hit Rate by 13.22% to 14.23% points, and NSS by 30.59 to 36.80 points. The largest improvements appear in NSS, suggesting that retrieval-grounded agentic reasoning is especially helpful for producing better process ordering and more complete treatment train structures.

The AeroNoxis variants also show smaller performance gaps across different backbones. While the standalone models vary noticeably in F1 and sequence quality, their AeroNoxis-enhanced versions achieve more comparable results. AeroNoxis-GPT-5.3 remains the best-performing system, reaching the highest precision, recall, F1, Hit Rate, and NSS. These results indicate that AeroNoxis is not merely improving a single foundation model, but provides a general framework for strengthening domain-specific treatment train design.

4.3 Ablation study

To examine the contribution of different knowledge components, we conduct an ablation study on APCTrainBench, as shown in Table 2. The multi-agent system without RAG achieves an average F1-score of 59.22%, indicating that agent collaboration alone is insufficient for reliable treatment train design. Introducing the unit-level knowledge base improves the average score to 64.58%, with consistent gains across easy, middle, and difficult scenarios. This shows that unit-level evidence helps the system better match pollutants with appropriate treatment technologies. Further adding template-level knowledge increases the average score to 69.46%, with a particularly clear improvement on difficult cases from 57.31% to 63.49%, suggesting that template knowledge provides useful process-level priors for arranging treatment units into feasible sequences.

Table 2: Ablation results on APCTrainBench comparing different knowledge configurations across three difficulty levels. The backbone is DeepSeek-V4 and metric is F1. Numbers in parentheses are standard errors for five independent runs.

Method	Easy	Middle	Difficult	Avg.
Multi-agent w/o RAG	$66.42_{(1.18)}$	$59.36_{(1.43)}$	$51.87_{(1.76)}$	$59.22_{(1.31)}$
Unit KB only	$71.58_{(0.96)}$	$64.84_{(1.22)}$	$57.31_{(1.54)}$	$64.58_{(1.17)}$
Unit + Template KB	$75.27_{(0.82)}$	$69.63_{(1.05)}$	$63.49_{(1.36)}$	$69.46_{(1.03)}$
HKB	$\mathbf{78.64}_{(0.71)}$	$\mathbf{74.28}_{(0.93)}$	$\mathbf{68.86}_{(1.18)}$	$\mathbf{73.93}_{(0.89)}$

The full HKB achieves the best performance across all difficulty levels, reaching 78.64%, 74.28%, and 68.86% on easy, middle, and difficult scenarios, respectively, with the highest average score of 73.93. Compared with the no-RAG variant, HKB improves the average score by 14.71 points, and the gains become larger as task difficulty increases: 12.22 points on easy cases, 14.92 points on middle cases, and 16.99 points on difficult cases. Meanwhile, the average standard deviation decreases from 1.31% to 0.89%, showing that HKB not only improves accuracy but also enhances stability. Overall, these results demonstrate that unit-level knowledge, template-level knowledge, and their hierarchical integration make complementary contributions to the robustness and engineering feasibility of AeroNoxis.

5 Conclusion

We propose AeroNoxis, a retrieval-grounded multi-agent framework for early-stage industrial waste-gas treatment train design. AeroNoxis addresses this need by turning treatment train generation into a staged reasoning process, where candidate designs are planned, checked, revised, and explained before being returned to the user. Experiments on APCTrainBench show that this workflow leads to more reliable treatment train recommendations than standalone LLMs and single-agent RAG baselines. The results suggest that retrieval alone is not sufficient, as domain evidence must be organized to support process-level reasoning and generated designs must be examined against explicit engineering rules. Ablation results further show that the hierarchical knowledge base and the critic agent play complementary roles, with the former improving the grounding and completeness of generated trains and the latter reducing serious violations. AeroNoxis is intended as a decision-support tool rather than an automated engineering designer. Its value lies in transparent early-stage screening, especially when users need to compare candidate control strategies under incomplete information. Future work will extend the knowledge base, strengthen cost and operability estimation, and connect the framework with more detailed engineering models. Its outputs should still be reviewed by domain experts before practical use.

References

[1] Hailin Wang, Shumei Sun, Lei Nie, Zhongshen Zhang, Wenpeng Li, and Zhengping Hao. A review of whole-process control of industrial volatile organic compounds in china. *Journal of Environmental Sciences*, 123:127–139, 2023.

[2] Cuiting Yang, Guang Miao, Yunhong Pi, Qibin Xia, Junliang Wu, Zhong Li, and Jing Xiao. Abatement of various types of vocs by adsorption/catalytic oxidation: A review. *Chemical Engineering Journal*, 370:1128–1153, 2019.

[3] Muhammad Shahzad Kamal, Shaikh A Razzak, and Mohammad M Hossain. Catalytic oxidation of volatile organic compounds (vocs)–a review. *Atmospheric Environment*, 140:117–134, 2016.

[4] Xin Wu, Yan Lin, Yongyi Wang, Shaohua Wu, and Chunping Yang. Volatile organic compound removal via biofiltration: influences, challenges, and strategies. *Chemical Engineering Journal*, 471:144420, 2023.

[5] Rong Chen, Tongsheng Zhang, Yiqun Guo, Jiawei Wang, Jiangxiong Wei, and Qijun Yu. Recent advances in simultaneous removal of so2 and nox from exhaust gases: Removal process, mechanism and kinetics. *Chemical Engineering Journal*, 420:127588, 2021.

[6] Damien Evrard, Jonathan Villot, Chadad Armiyaou, Rodolphe Gaucher, Sofia Bouhrizi, and Valérie Laforest. Best available techniques: An integrated method for multicriteria assessment of reference installations. *Journal of Cleaner Production*, 176:1034–1044, 2018.

[7] Mohsen Tayebi, Akram Bemani, Abdolvahhab Fetanat, and Masoud Fehresti-Sani. A decision support system for sustainability prioritization of air pollution control technologies in energy and carbon management: Oil & gas industry of iran. *Journal of Natural Gas Science and Engineering*, 99:104416, 2022.

[8] Wenhao An, Jun Zhao, Jiangang Lu, Yi Han, and Dong Li. Zero-liquid discharge technologies for desulfurization wastewater: A review. *Journal of environmental management*, 321:115953, 2022.

[9] Hongwei Ren, Liang Han, Qing Li, Xiaoyun Liu, Xiaoyu Huang, Luyu Jia, Zhaoxiong Huang, and Jinfeng Zhang. Enhancing the thermal stability of activated carbon in vocs adsorption by incorporation of silicon-doped cqds. *Journal of Environmental Chemical Engineering*, 13(3):117114, 2025.

[10] Feng Han, Yirong Deng, Qiyuan Liu, Yongzhang Zhou, Jun Wang, Yongjian Huang, Qianlong Zhang, and Jing Bian. Construction and application of the knowledge graph method in management of soil pollution in contaminated sites: A case study in south china. *Journal of environmental management*, 319:115685, 2022.

[11] Yadagiri Annepaka and Partha Pakray. Large language models: a survey of their development, capabilities, and applications. *Knowledge and Information Systems*, 67(3):2967–3022, 2025.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[13] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39, 2024.

[14] Filippo Chiarello, Simone Barandoni, Marija Majda Škec, and Gualtiero Fantoni. Generative large language models in engineering design: opportunities and challenges. *Proceedings of the Design Society*, 4:1959–1968, 2024.

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[16] Yizheng Huang and Jimmy Xiangji Huang. A survey on retrieval-augmented text generation for large language models. *ACM Computing Surveys*, 2024.

[17] Tian-Yi Che, Xian-Ling Mao, Tian Lan, and Heyan Huang. A hierarchical context augmentation method to improve retrieval-augmented llms on scientific papers. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 243–254, 2024.

[18] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[19] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

[20] Qiyang Nie and Tong Liu. Large language models for environmental modeling: Framework, capabilities, constraints. *Journal of Environmental Management*, 398:128417, 2026.

[21] Jun Song, Chendong Ma, and Maohao Ran. Airgpt: pioneering the convergence of conversational ai with atmospheric science. *Npj Climate and Atmospheric Science*, 8(1):179, 2025.

[22] Jiashu Ye, Tong Wu, Weiwen Chen, Hao Zhang, Zeteng Lin, Xingxing Li, Shujuan Weng, Manni Zhu, Xin Yuan, Xinlong Hong, et al. Emission-gpt: A domain-specific language model agent for knowledge retrieval, emission inventory and data analysis. *arXiv preprint arXiv:2510.02359*, 2025.

[23] Sabri Ghazi, Julie Dugdale, and Tarek Khadir. Modelling pm10 crisis peaks using multi-agent based simulation: Application to annaba city, north-east algeria. In *International Conference on Information Systems for Crisis Response and Management in Mediterranean Countries*, pages 65–72. Springer, 2015.

[24] Sabri Ghazi, Julie Dugdale, and Tarek Khadir. Modelling air pollution crises using multi-agent simulation. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 172–177. IEEE, 2016.

[25] Sabri Ghazi, Julie Dugdale, and Tarek Khadir. A multi-agent based approach for simulating the impact of human behaviours on air pollution. *arXiv preprint arXiv:1904.05429*, 2019.

[26] Yu-Erh Pan and Ayesha Siddika Nipu. Llm-enhanced air quality monitoring interface via model context protocol. In *2025 7th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pages 1–6. IEEE, 2025.

[27] Ichrak Mokhtari, Walid Bechkit, Mohamed Sami Assenine, and Hervé Rivano. Navigating the smog: a cooperative multi-agent rl for accurate air pollution mapping through data assimilation. *arXiv preprint arXiv:2407.12539*, 2024.

[28] Yangxinyu Xie, Bowen Jiang, Tanwi Mallick, Joshua David Bergerson, John K Hutchison, Duane R Verner, Jordan Branham, M Ross Alexander, Robert B Ross, Yan Feng, et al. A rag-based multi-agent llm system for natural hazard resilience and adaptation. *arXiv preprint arXiv:2504.17200*, 2025.

[29] Yangxinyu Xie, Bowen Jiang, Tanwi Mallick, Joshua David Bergerson, John K Hutchison, Duane R Verner, Jordan Branham, M Ross Alexander, Robert B Ross, Yan Feng, et al. Wildfiregpt: Tailored large language model for wildfire analysis. *arXiv preprint arXiv:2402.07877*, 2024.

[30] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.

[31] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[32] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[33] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[34] DeepSeek-AI. Deepseek-v4 technical report, 2024.

[35] Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.